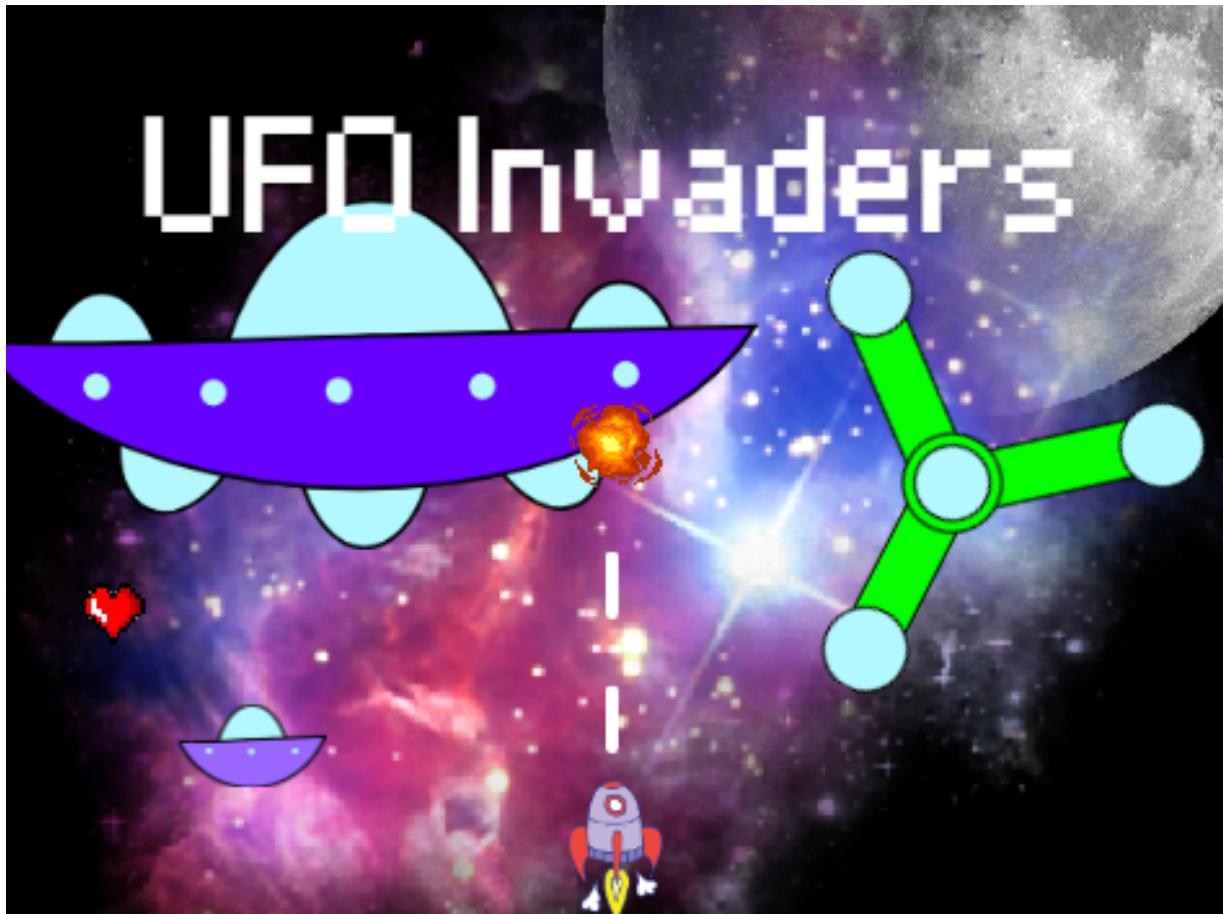


UFO Invaders

Programming in Scratch

<https://scratch.mit.edu/projects/967248315>

Victor McClure



Contents

Introduction	3
Accomplishments	3
Skills Learned	3
Design Ideas	4
Budget	4
Summary of What Was Done	5
Development Process and Final Project	5
Code for the Rocket	12
Code for the Lasers	17
Code for the UFOs	19
Code for the Health Bars	23
Code for the Explosions	24
Code for the Stars	26
Code for the Nebulae	28
Code for the Score	30
Code for the Boss	32
Code for the Enemy Laser	42
Code for the Enemy Spawn	44
Code for the Earth	45
Code for the Planets	47
Code for the Power-Ups	48
Code for the Play Button	49
Code for the Title	50
Code for the Game Over	51
Code for the Victory	53
Code for the Death Ray	54
Code for the Background	55
References	56

For my project, I coded a game in a free block-based program called Scratch. In my game, you fly to space in a rocket and shoot UFOs. It has different types of UFOs, boss UFOs at the end of each level, and items to collect. The heart item gives you extra life, and the wand gives you 3 lasers. Shields make you more resistant against UFOs. You use the arrow keys to move, and press space to shoot.

Accomplishments

While working on this project I finished multiple courses on Udemy, a website where you watch videos and learn things. I finished a Udemy course called Advanced Scratch Programming for Kids by Daniel Ciocirlan. Since then I have created more than 100 Scratch projects. I also started teaching coding to kids. First, I was teaching coding to my friends at the library. Also, I had a Boy Scout requirement to make some money, so I started teaching kids coding for profit. I posted a flyer on Facebook, and I found a few customers. I love coding because I can make any game I can imagine!

Skills Learned

While coding UFO Invaders, I learned how to make high-quality games like Mario and Space Invaders. I also learned how to make power-ups, lives, intelligent enemies, sound loops, visual effects, levels, boss fights, explosions, lasers, and more! I learned creativity and abstract thinking because I had to plan and imagine the game. I also learned logic and problem-solving by debugging my mistakes in the code. I learned resilience because I did not give up while coding this game, and finished the whole project.

Some of the coding skills I learned are: sequence, iteration, conditionals, variables, loops, user input, functions and lists.

I used sequencing to put the blocks in the correct order. I used iteration to make the rocket move. Conditionals were used to do if-statements like if the space key is pressed, then shoot. Variables like "life" were used to calculate the health, and to see if the rocket's health is 0 and if it should explode. I used loops to repeat things like spawning enemies. User input was used when a key is pressed to do things, like moving and shooting. I used the move x and move y functions to move the player. Lists were used to hold the positions of the health bars, lasers, power-ups and explosions.

Design Ideas

At first I thought about making a raining taco game, but decided it was too simplistic and I couldn't come up with a good idea of how to play it. Next I decided to make a game about Space because I always loved Space. I played Space Invaders before so I decided to make my own version of the game. In my game you fly up, but in Space Invaders, you can only move left and right and there are more levels in my game. My game also has better graphics.

Budget

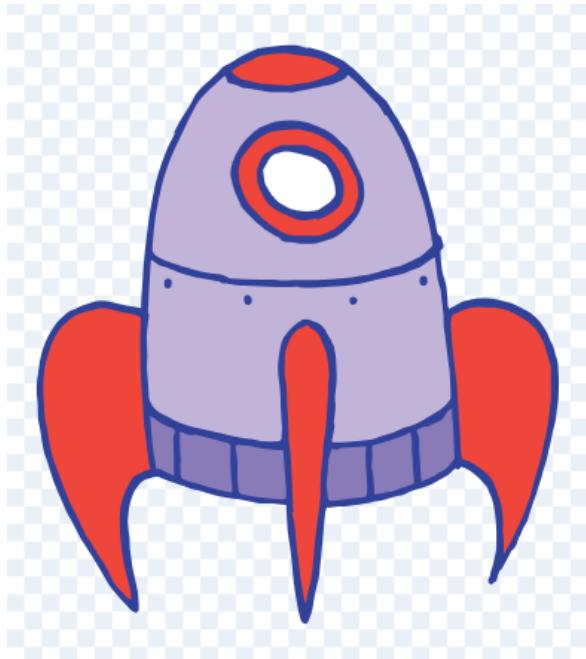
The only money I spent on this project was the \$10 Udemy course called Advanced Scratch Programming for Kids by Daniel Ciocirlan, that I took to learn coding. I used my computer and internet that I already had. Scratch is the free program that I used to make this game.

Summary of What Was Done

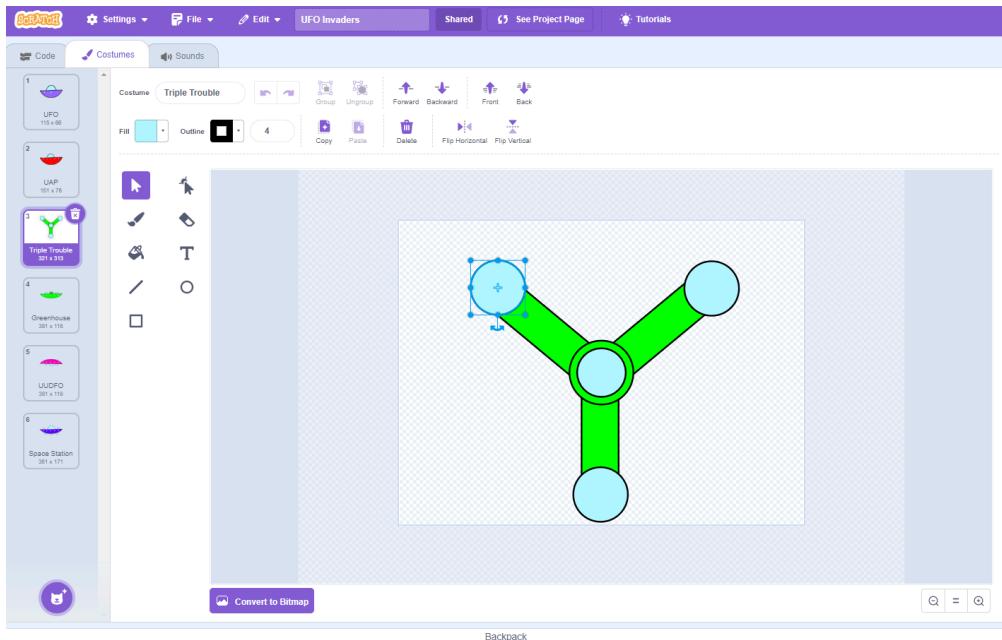
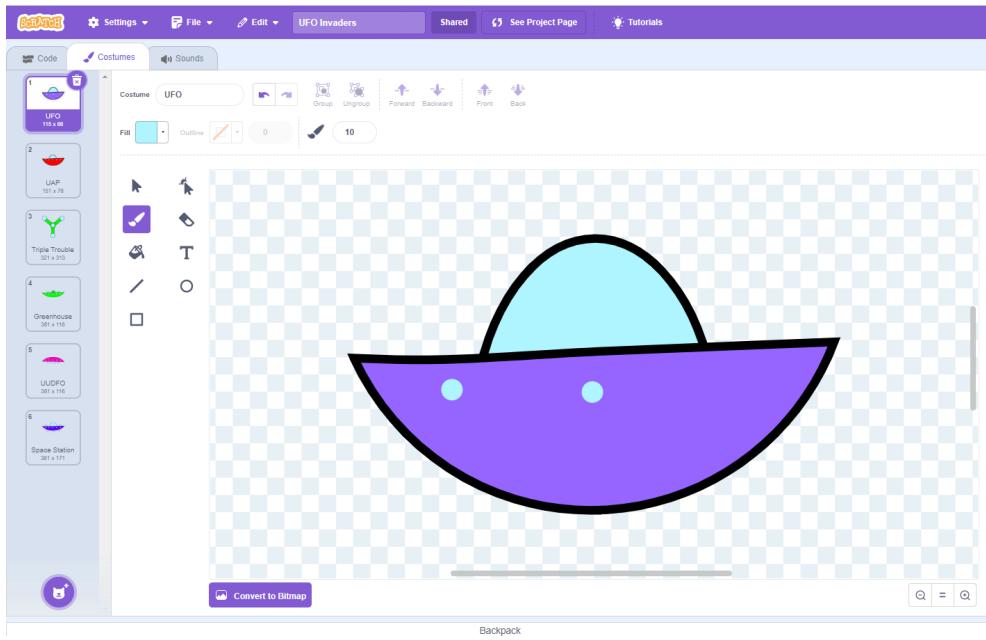
In this game I used Scratch to code a game about shooting UFOs. I learned how to code advanced Scratch games with a Udemy course. In this project, you fly around in a rocket and shoot UFOs. There are different power-ups you can collect to get different abilities. The UFOs get harder and harder as you go. There are 7 levels and 7 boss UFOs. The health bar above your rocket shows how much health you have. If your health bar is low, then if you collect a heart, your health will go up. The wand gives you the ability to shoot 3 lasers at once.

Development Process and Final Project

I used a built-in graphic from Scratch like the rocket.



Some of the sprites I drew myself, like the UFOs.



And some of them I downloaded from the internet.

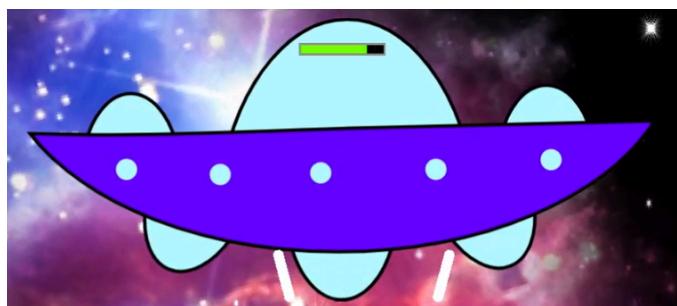


At first, the boss UFOs I drew myself but then I changed them because they looked too much like regular UFOs.



Half of the boss UFOs I got from the Udemy course called Advanced Scratch Programming for Kids by Daniel Ciocirlan, and the other half I made. The final boss picture I downloaded from the internet. The final boss summons minions and shoots death rays to make the final battle more difficult. The minions are helpers to the final boss and can jump down at the player.

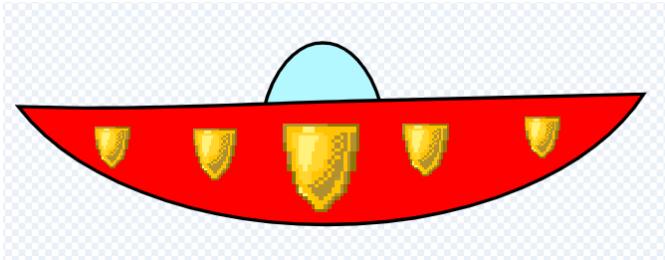
Space Station



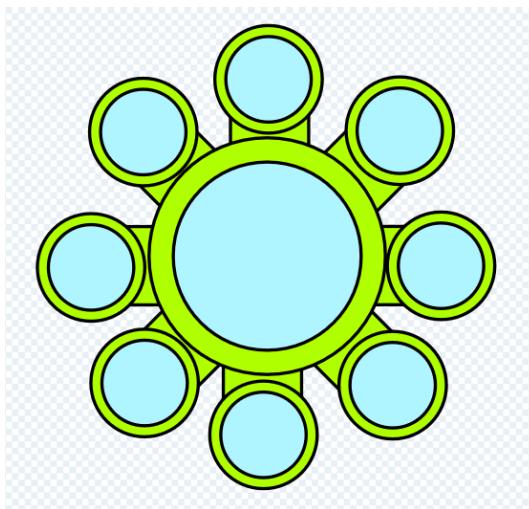
Moon Guardian



Shield



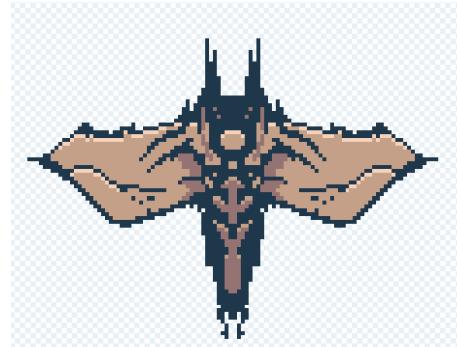
Octo-Sphere



Final Boss



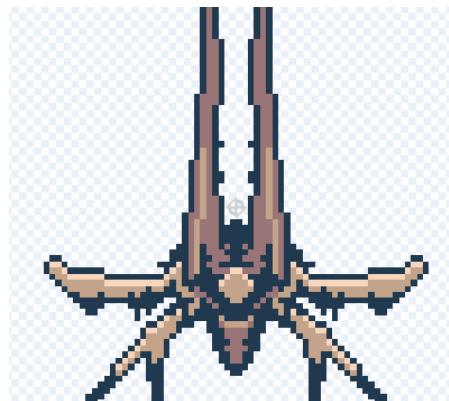
Mars Guardian



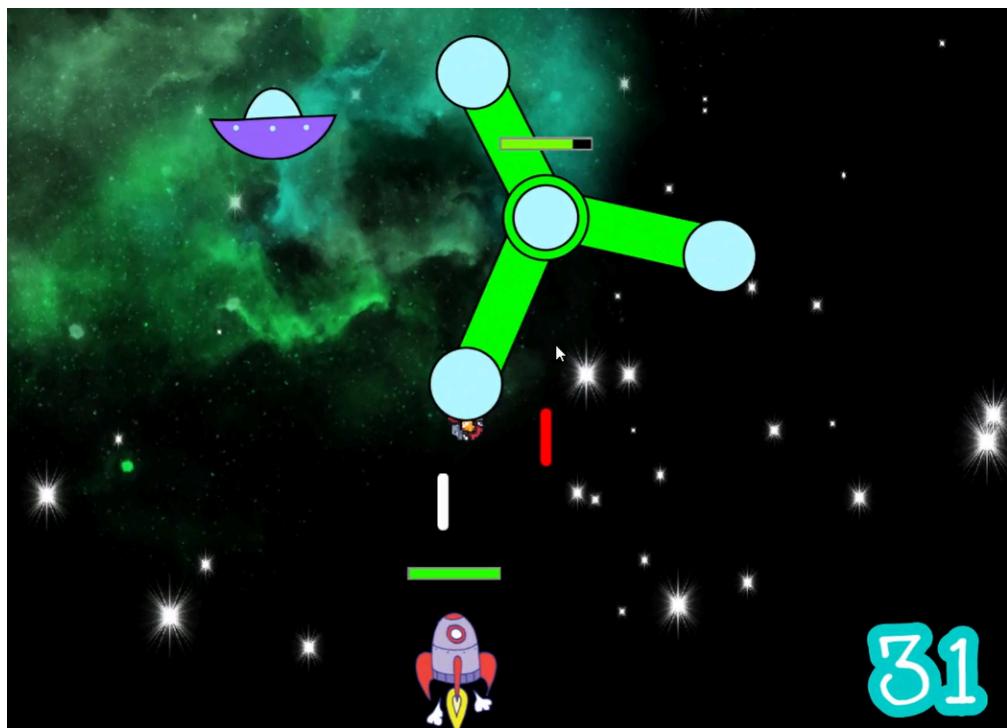
Boss Guardian



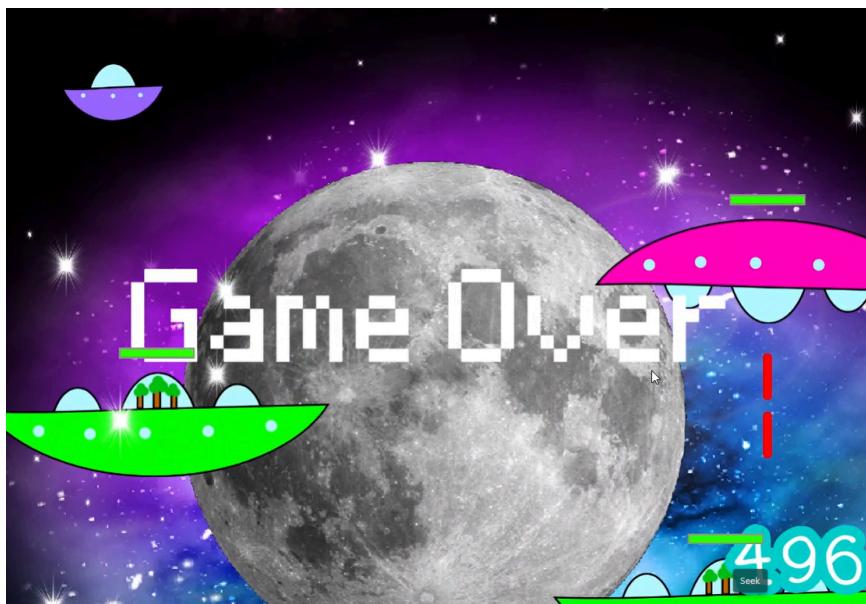
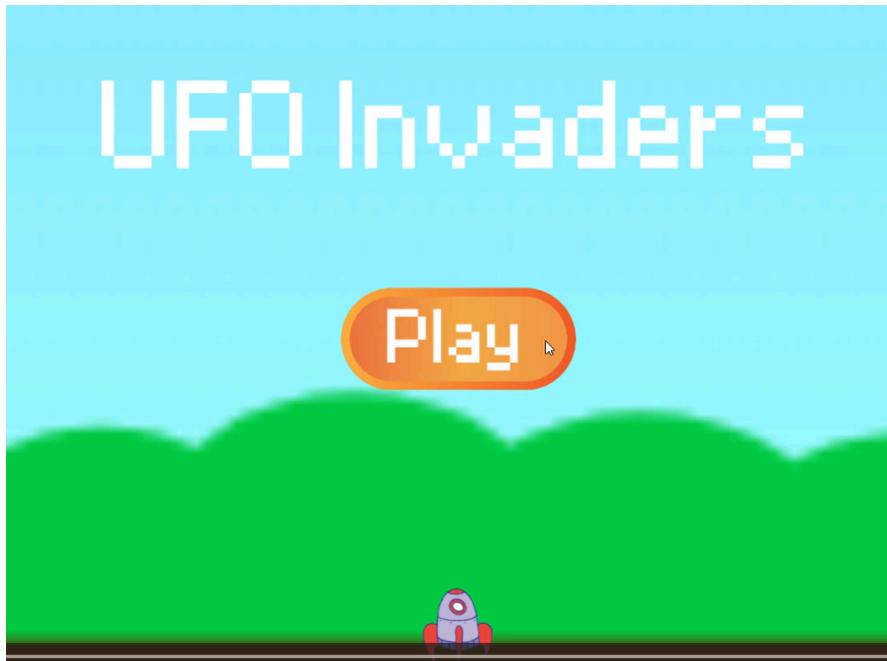
Minion



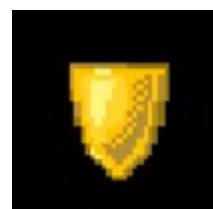
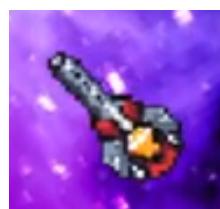
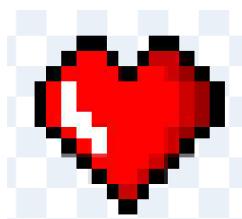
This game has 7 levels and 7 bosses. Each level gets harder and harder.



In the start screen there is a play button and a title. When you press the play button the game starts and the rocket lifts off. When you die you see a game over screen.



I've included multiple power-ups in this game. The hearts give you extra life if your health bar is low. The wand power-up grants you the ability to fire 3 lasers at once, and the shield makes you more resistant to UFOs and lasers.



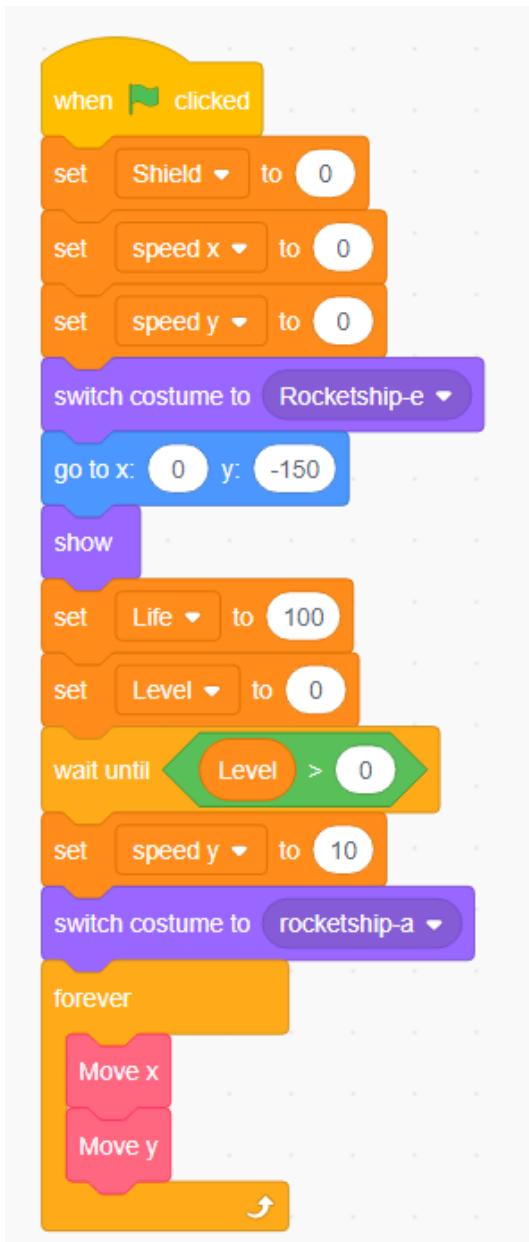
When you defeat the final boss you get a victory message.



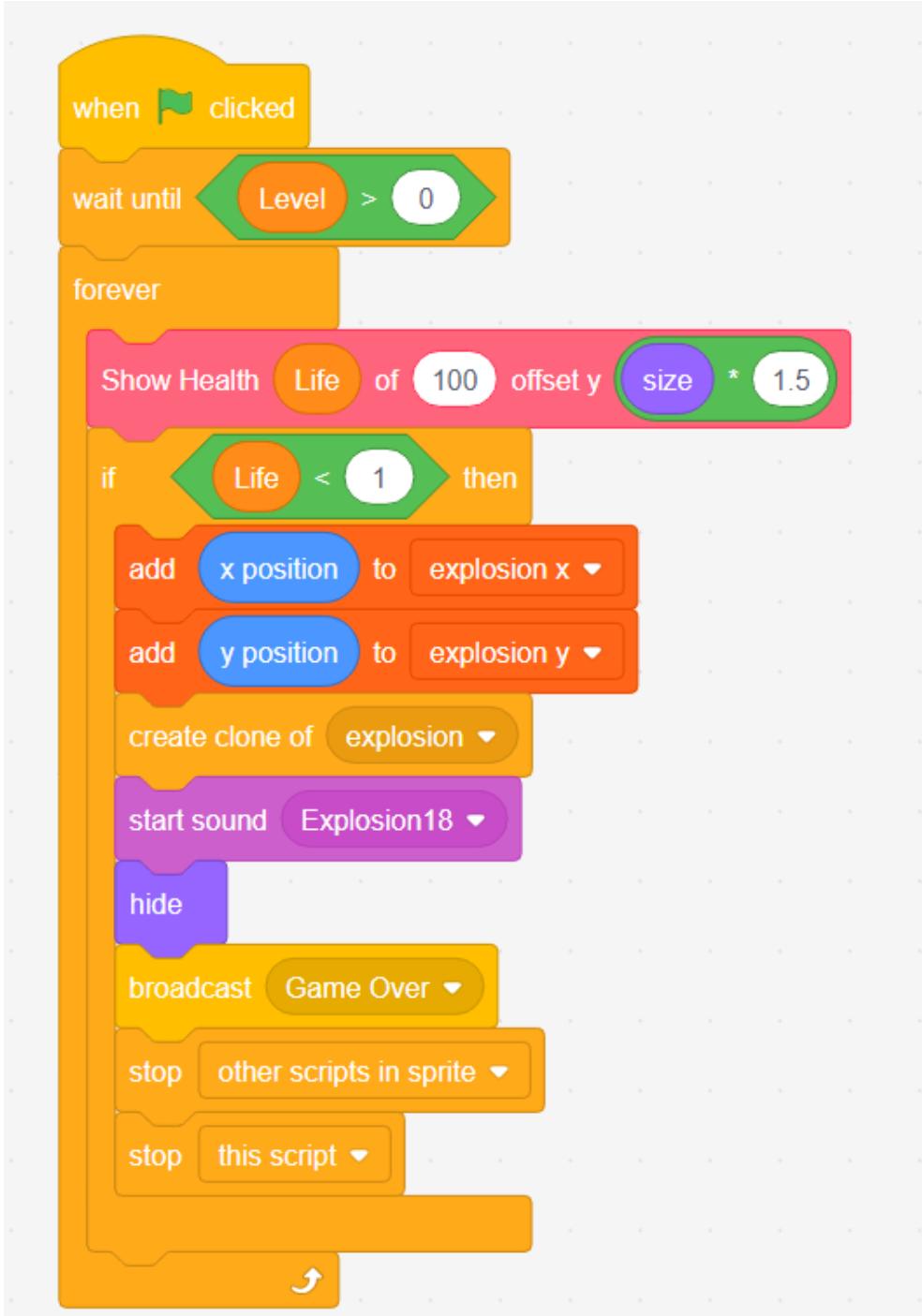
The Code For UFO Invaders

Code For The Rocket

This code resets the rocket at the start:



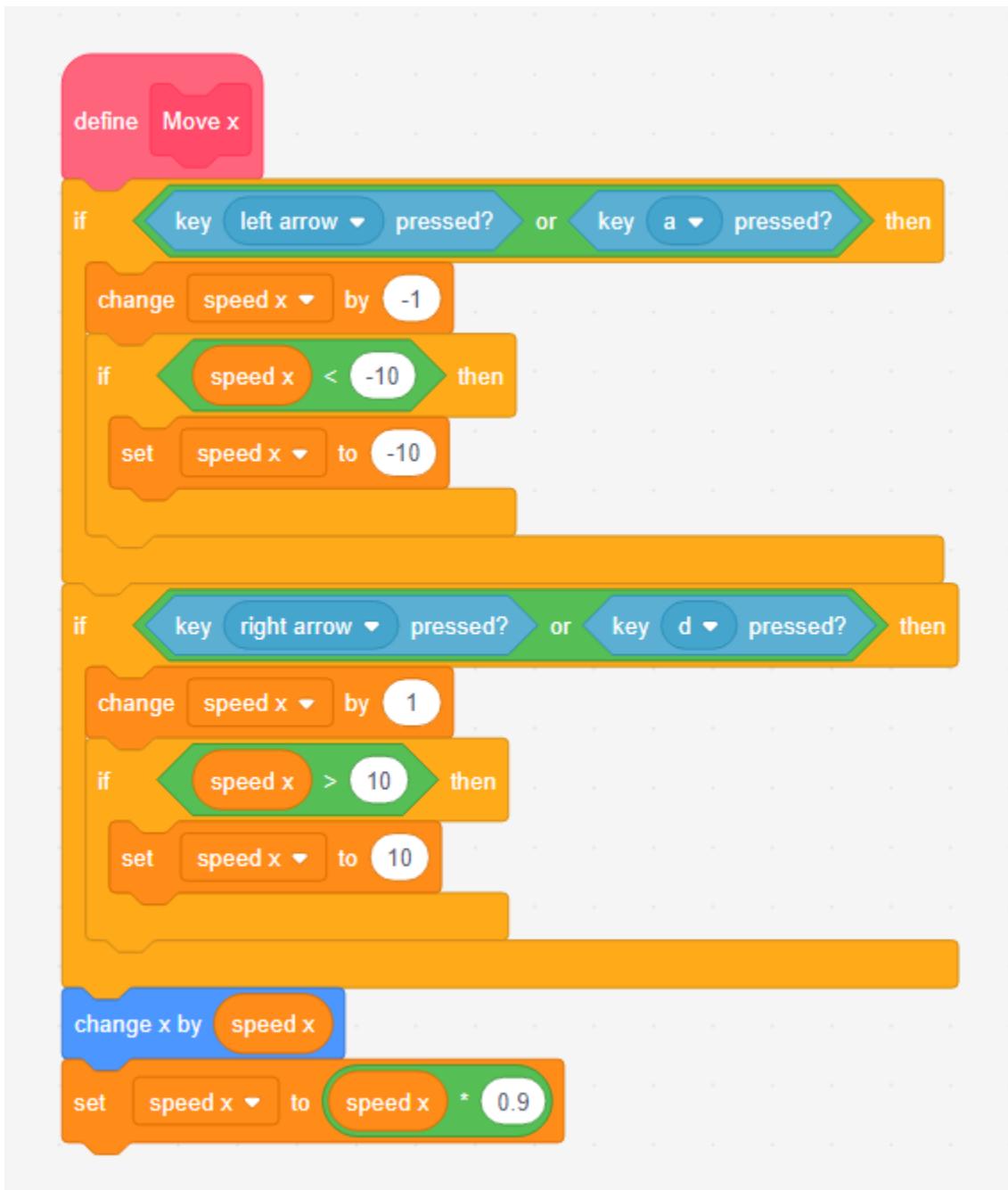
This code makes the rocket explode if the life is less than 1:



This code makes the rocket get more life when it gets the powerup, and makes the rocket more resistant when it gets the shield:



This code makes the rocket move horizontally:

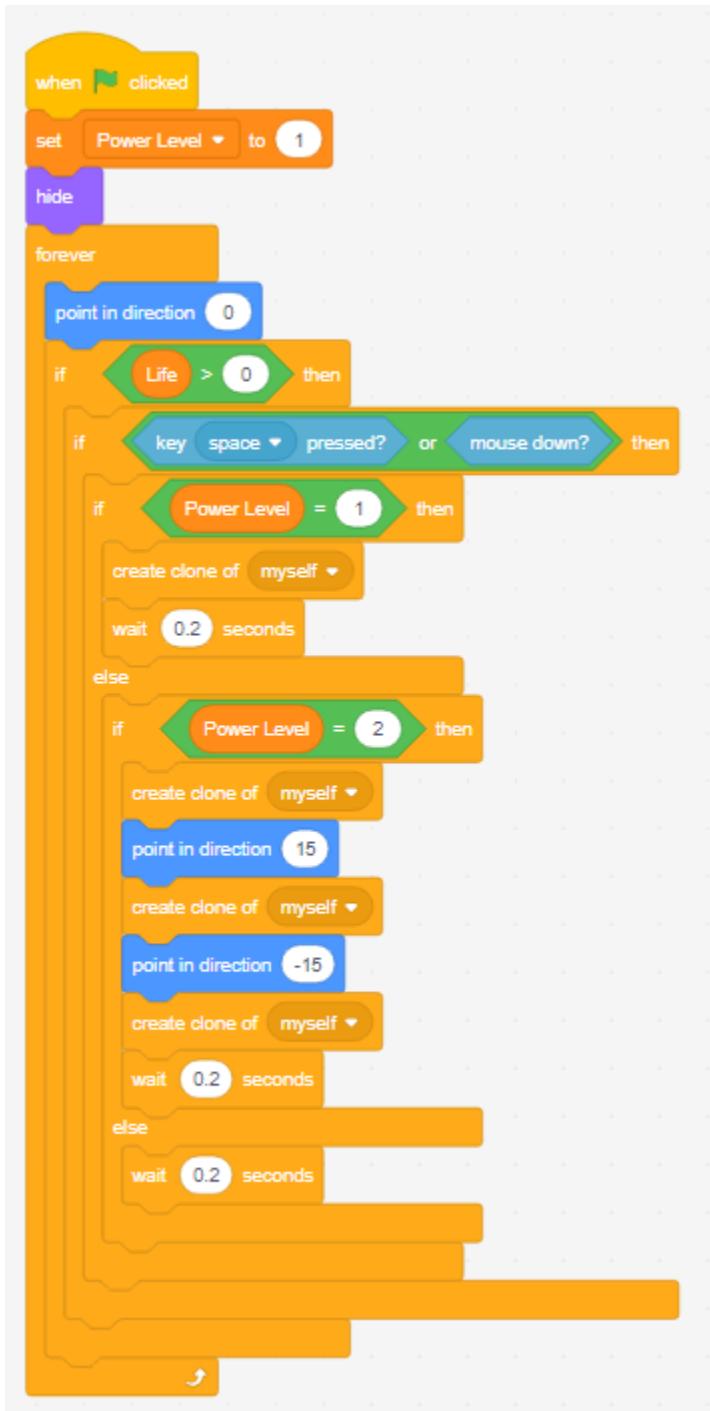


This code makes the rocket move vertically:

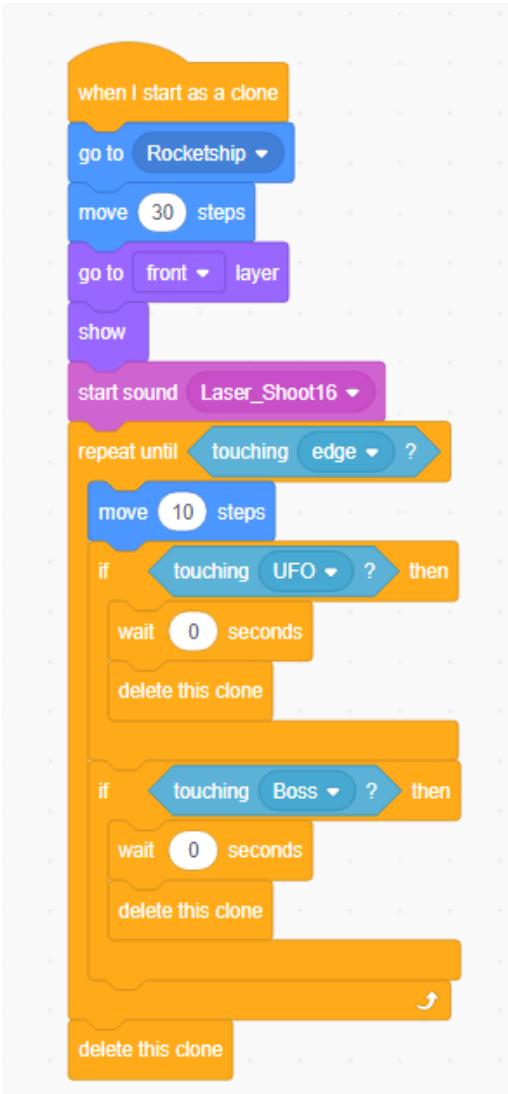


Code For The Lasers

This code fires the lasers:



This code makes the laser go up:

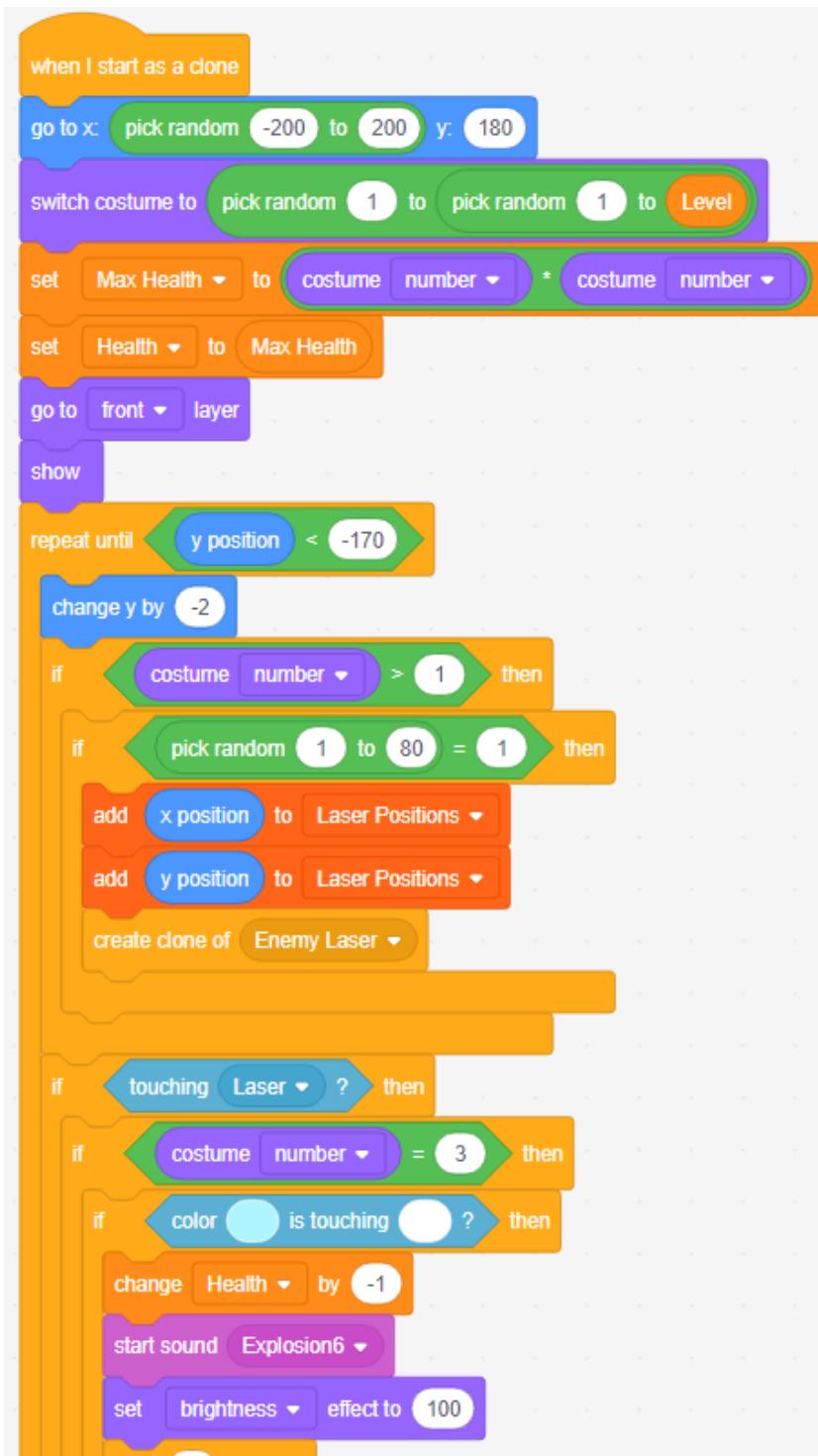


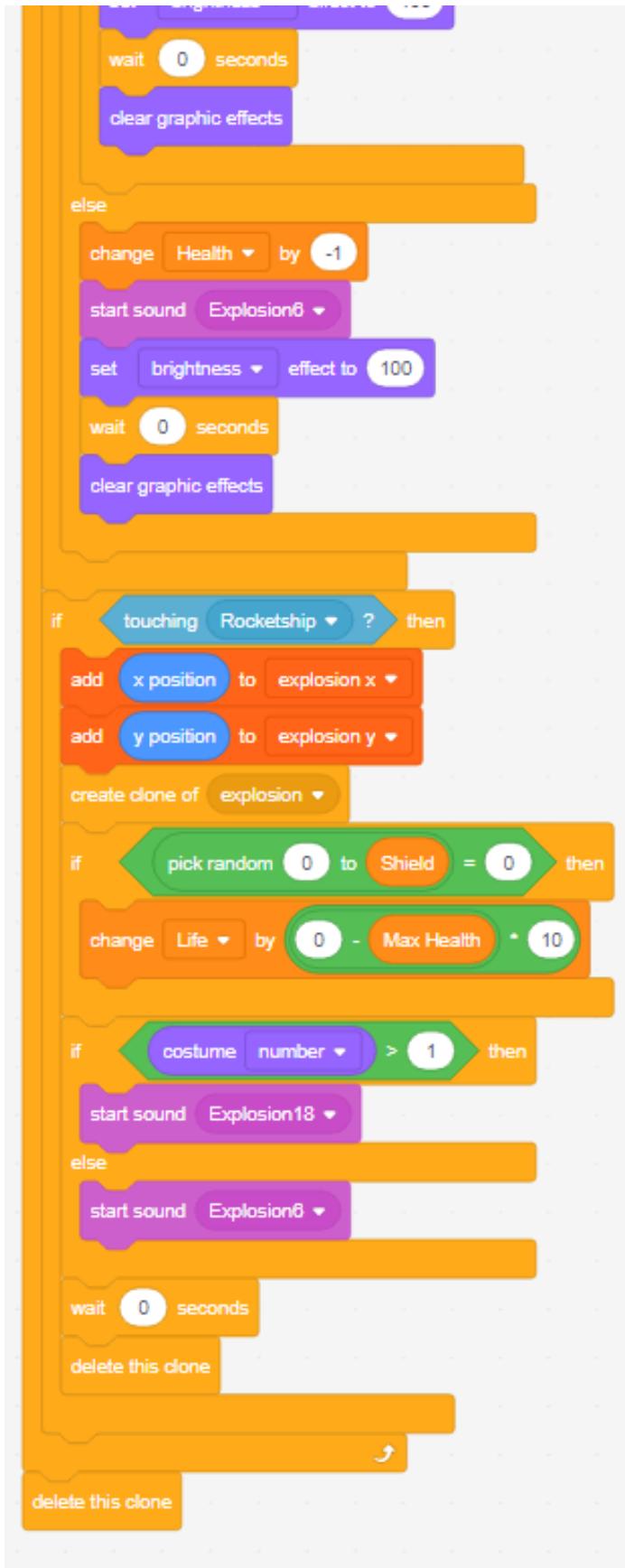
This makes the lasers fire more rapidly when the rocket gets a powerup:



Code For The UFOs

This code makes the UFO move and get shot by lasers if touching it:





This Scratch script is designed for an explosion clone. It starts with a `when green flag clicked` hat block containing:

- `wait [0 seconds]`
- `clear graphic effects`

Followed by an `else` branch:

- `change [Health v] by [-1]`
- `start sound [Explosion8 v]`
- `set [brightness v] effect to [100]`
- `wait [0 seconds]`
- `clear graphic effects`

Then, it includes an `if [touching [Rocketship v] ? then` control block:

- `add [x position] to [explosion x v]`
- `add [y position] to [explosion y v]`
- `create clone of [explosion v]`

For each created clone, it runs an `if [pick random 0 to [Shield] = 0 then` control block:

- `change [Life v] by [0 - Max Health * 10]`

It also includes an `if [costume number > 1 then` control block:

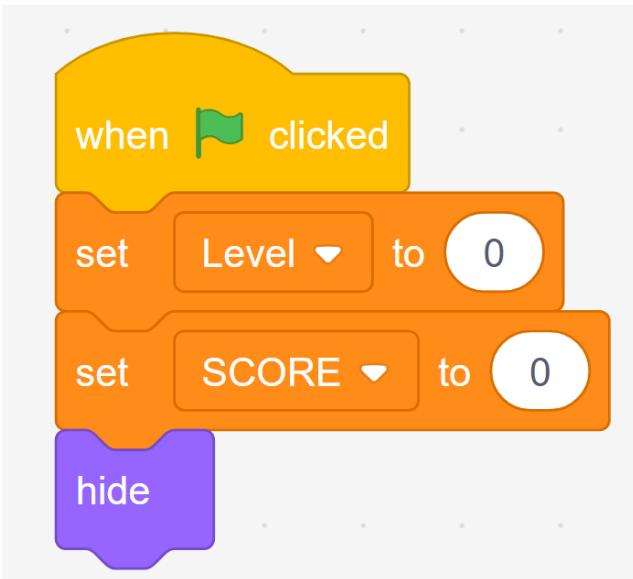
- `start sound [Explosion18 v]`

Otherwise, it runs:

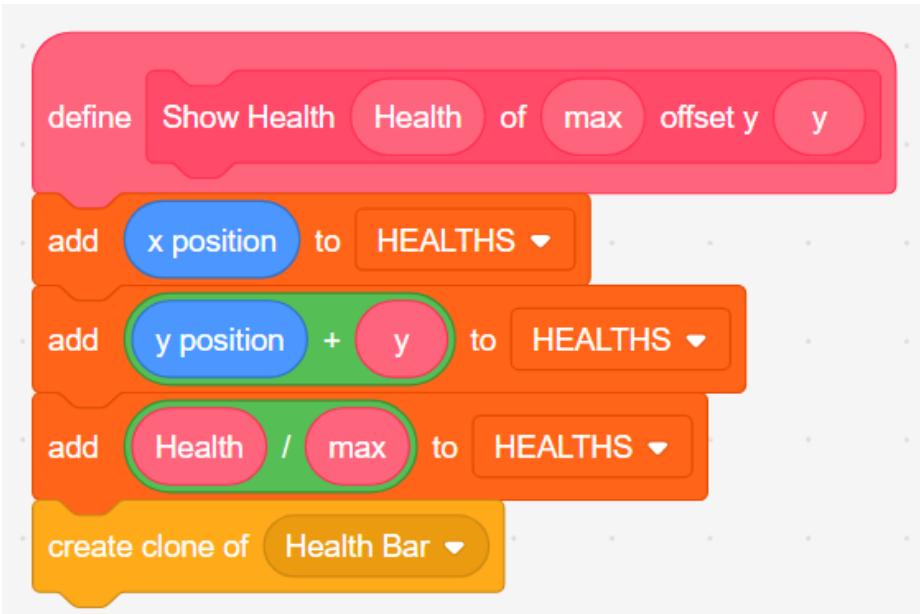
- `start sound [Explosion8 v]`
- `wait [0 seconds]`
- `delete this clone`

The script concludes with a `delete this clone` control block.

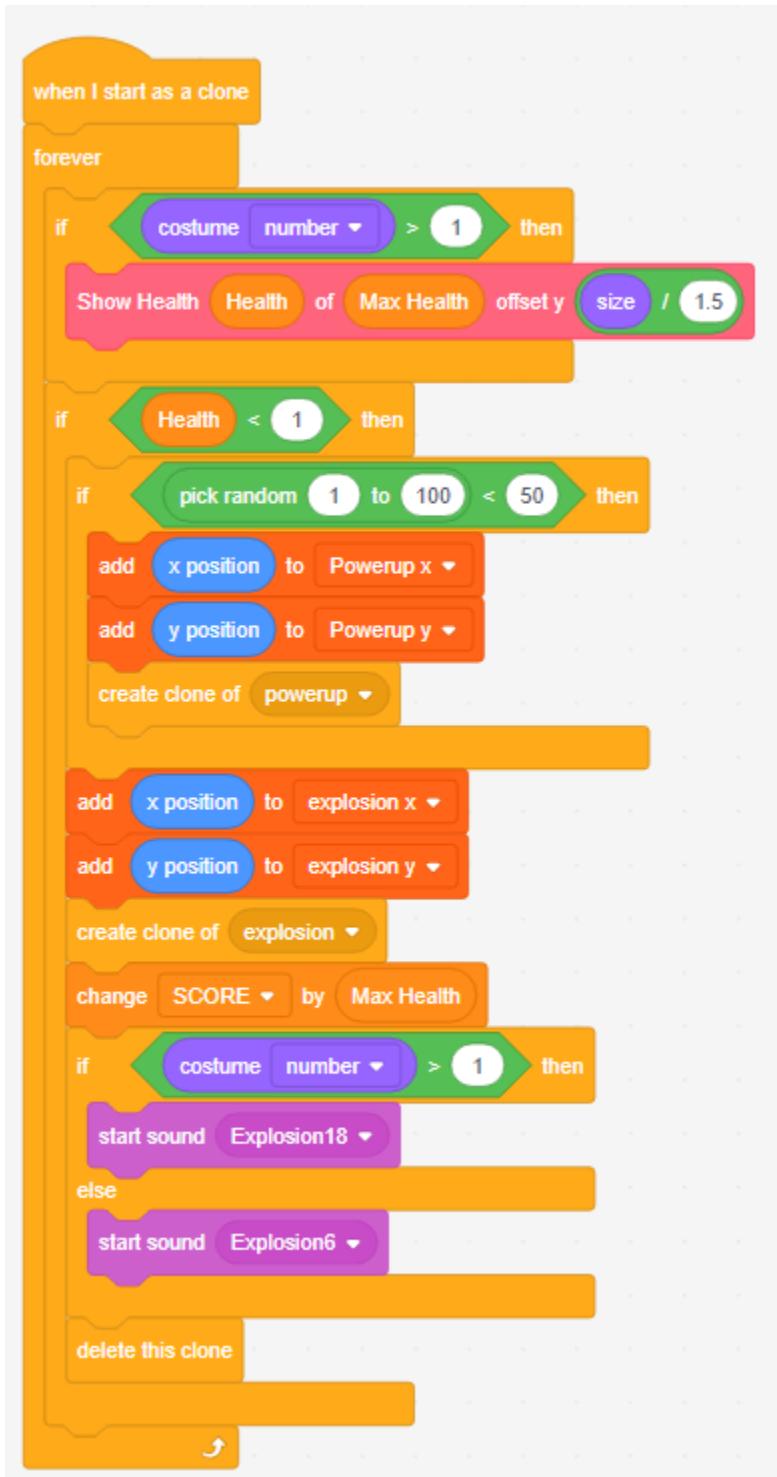
This resets the level and score:



This code makes the health bar show:



This code makes the UFO explode if the health is less than 1:



The Scratch script starts with the "when I start as a clone" hat. It enters a "forever" loop. Inside the loop, it checks if the costume number is greater than 1. If so, it shows the "Health" costume, which is set to the value of "Max Health" with an offset y of size / 1.5. Then, it checks if the "Health" variable is less than 1. If it is, it enters another if block where it picks a random number between 1 and 100, and if it's less than 50, it creates a clone of the "powerup" costume at a random position. This clone's x position is added to the "Powerup x" variable, and its y position is added to the "Powerup y" variable. After creating the powerup clone, it creates a clone of the "explosion" costume at the same coordinates. The explosion clone's x position is added to the "explosion x" variable, and its y position is added to the "explosion y" variable. Finally, it changes the "SCORE" by the value of "Max Health". It then checks if the costume number is greater than 1 again. If it is, it starts the sound "Explosion18". If not, it starts the sound "Explosion6". Finally, it deletes the current clone.

```
when I start as a clone
forever
  if [costume number v] > [1]
    then
      Show Health [Health v] of [Max Health v] offset y [size v / 1.5]
    if [Health v] < [1]
      then
        if [pick random (1) to (100) < 50]
          then
            add [x position v] to [Powerup x v]
            add [y position v] to [Powerup y v]
            create clone of [powerup v]
            add [x position v] to [explosion x v]
            add [y position v] to [explosion y v]
            create clone of [explosion v]
            change [SCORE v] by [Max Health v]
            if [costume number v] > [1]
              then
                start sound [Explosion18 v]
            else
              start sound [Explosion6 v]
            delete this clone
        end
      end
    end
  end
end
```

This code makes the UFO spin if it is the right type of UFO:

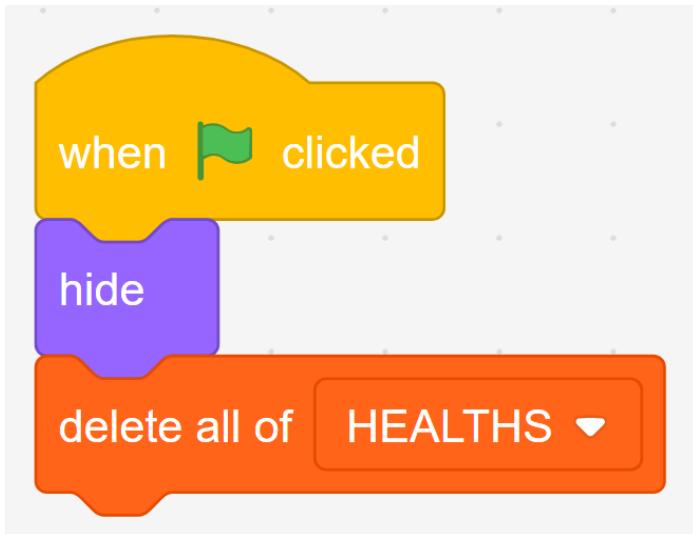
```
when I start as a clone
forever
  if [costume number] = 3 then
    turn (3) degrees
```

Code For The Health Bars

This code makes the health bar show:

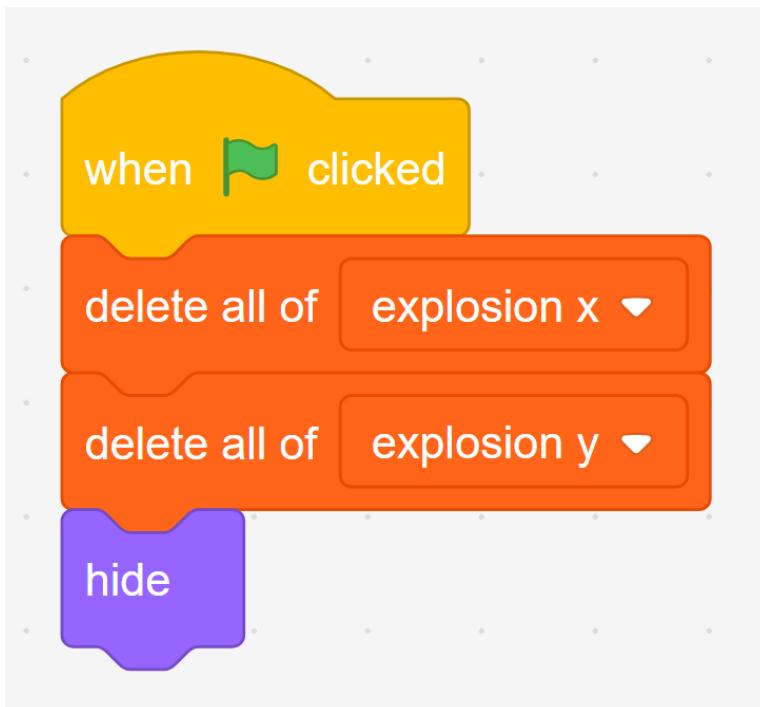
```
when I start as a clone
set x to [item 1 of HEALTHS v]
set y to [item 2 of HEALTHS v]
switch costume to [10 * item 3 of HEALTHS v]
delete [1] of HEALTHS
delete [1] of HEALTHS
delete [1] of HEALTHS
go to [front v] layer
show
if [touching edge?] then
  delete this clone
else
  wait [0] seconds
  delete this clone
```

This starts the health bars:

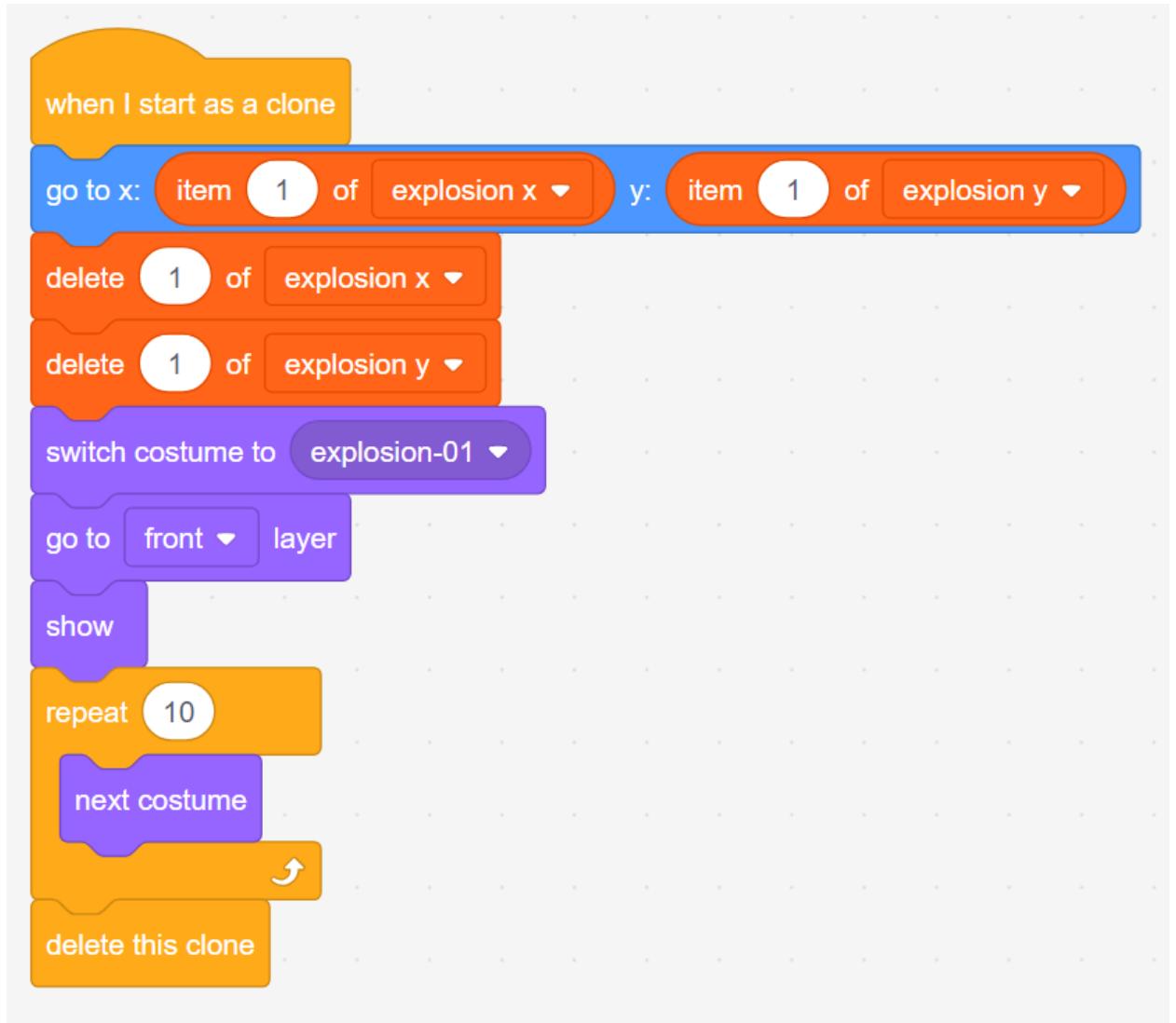


Code For The Explosions

This is the starting code for the explosion:

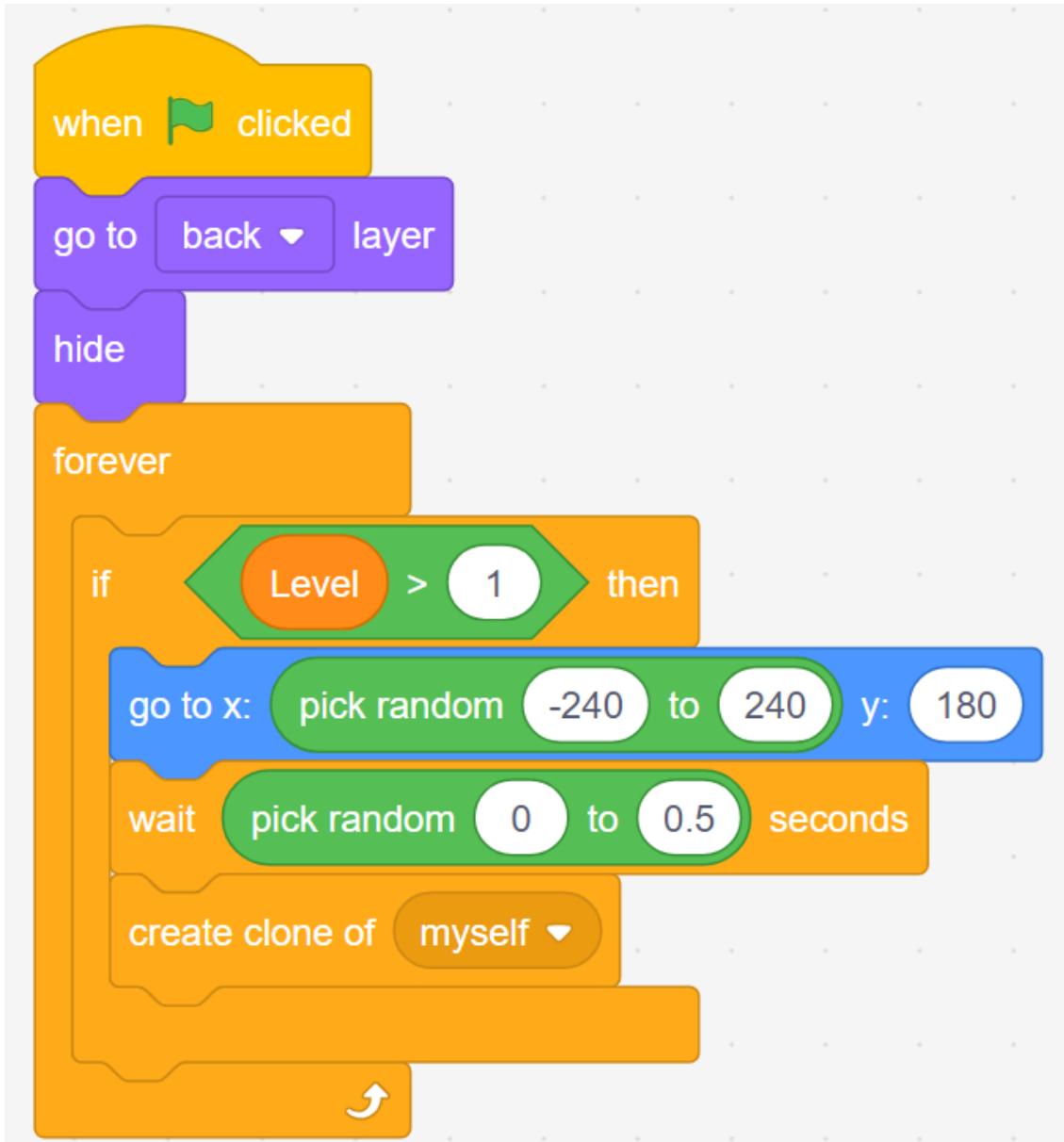


This code makes the explosion show:

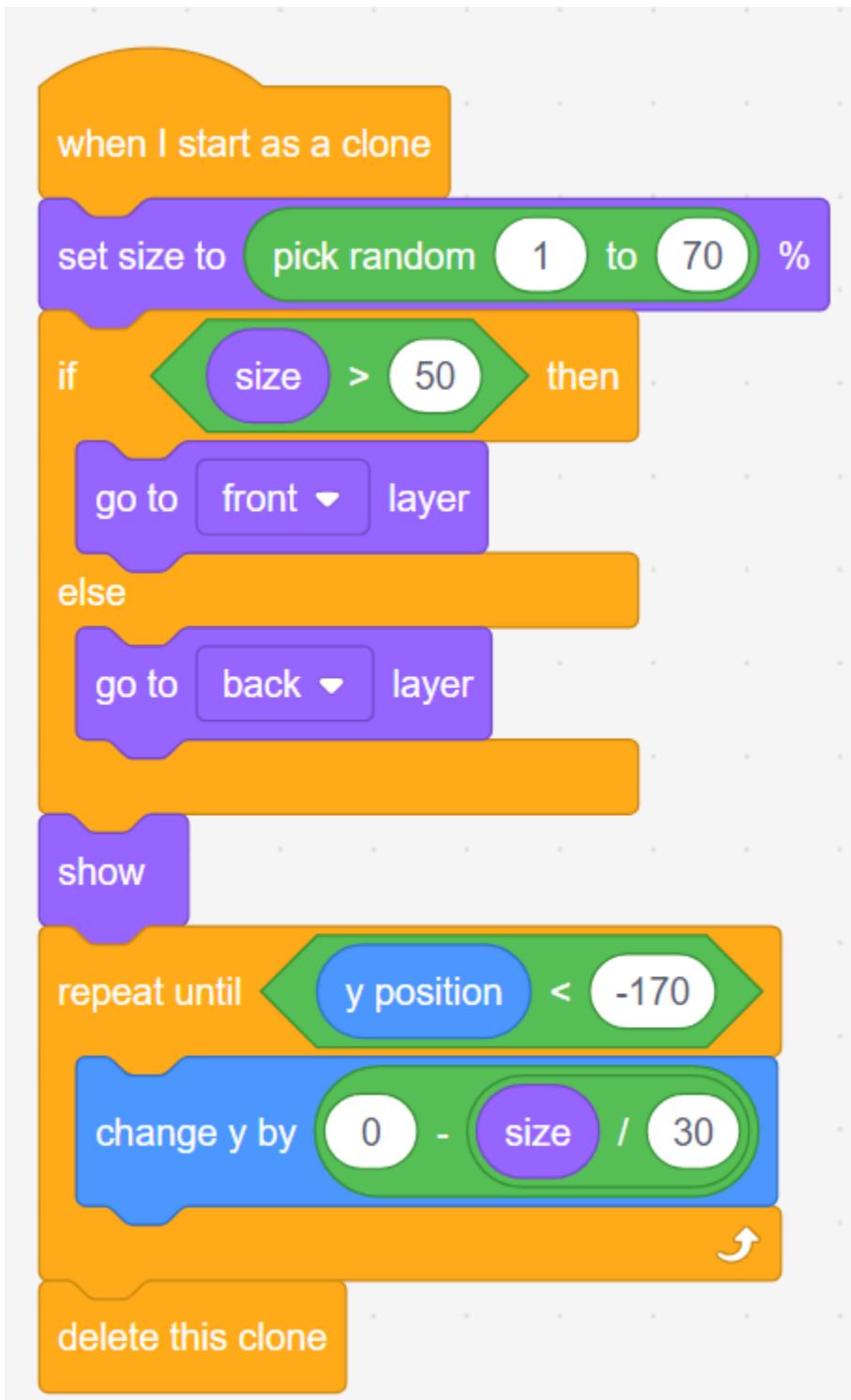


Code For The Stars

This code makes the stars appear:

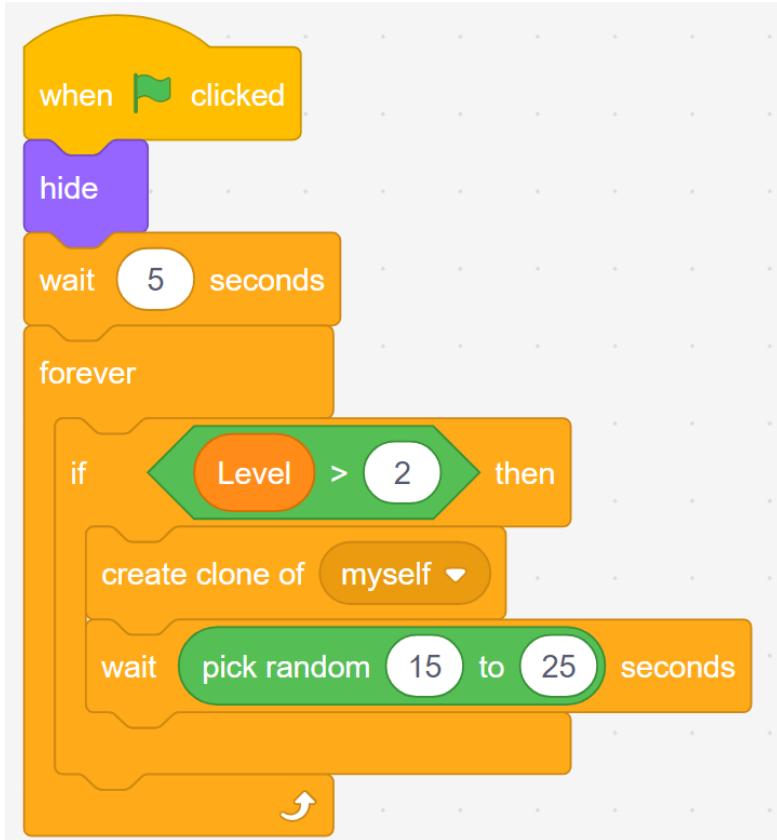


This code makes the stars move down:

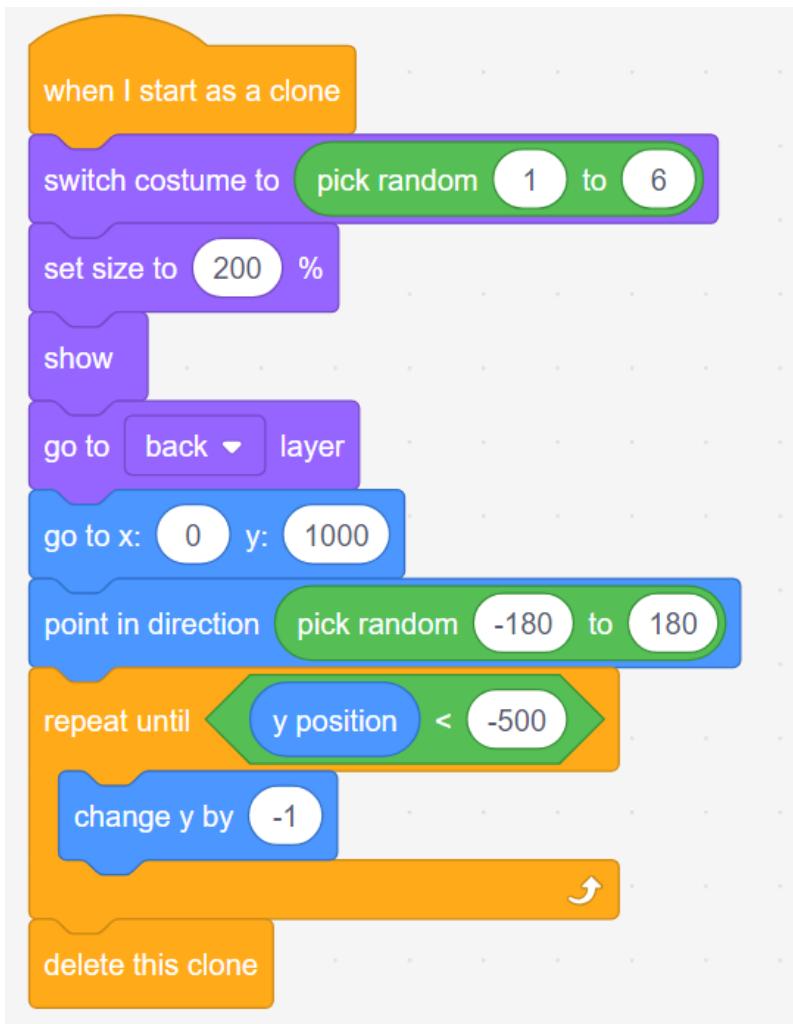


Code For The Nebulae

This code makes the nebulae appear:

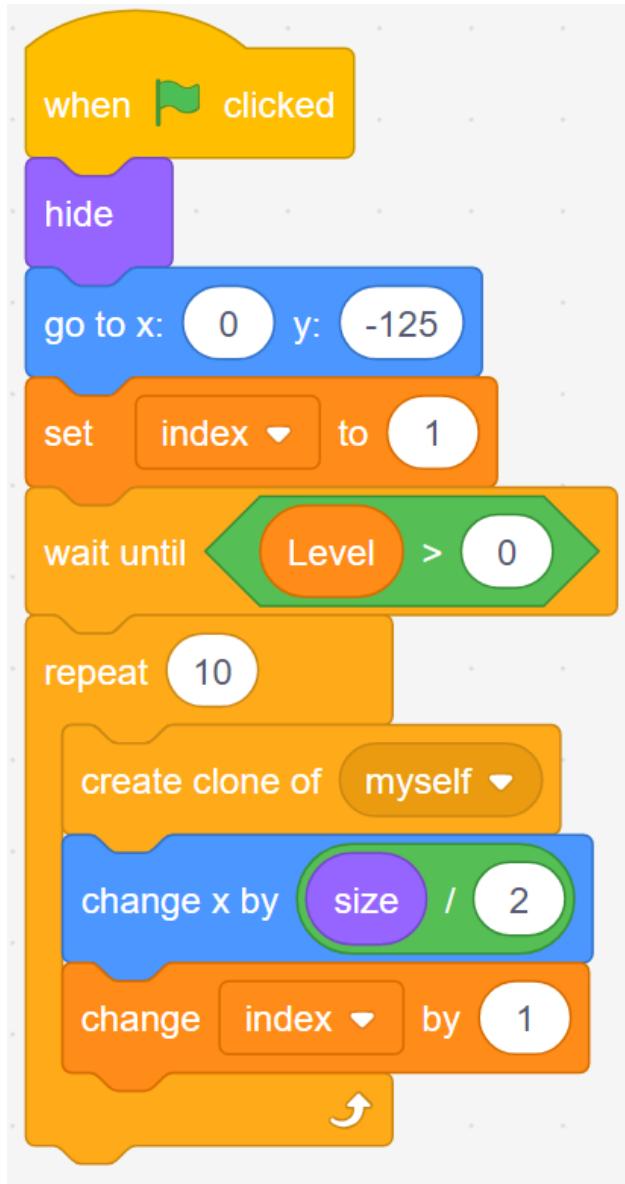


This code make the nebulae move:



Code For The Score

This code makes the score numbers:



This code makes the number show:

```
when I start as a clone
show
forever
  go to front layer
  change costume to letter index of SCORE
  set x to (index - 0.5) - (length of SCORE / 2) * (size / 2) + 200
  change size by (0.2 * 50) - size
```

The Scratch script starts with a "when I start as a clone" hat block. It then has a "show" control block. The main loop begins with a "forever" control block. Inside the loop, there is a "go to front layer" control block, followed by a "change costume to" block which uses the "letter" and "index" parameters of the "SCORE" variable. Below it is a "set x to" block that calculates the horizontal position based on the index of the "SCORE" variable, scaled down by 0.5 and centered around 200. Finally, there is a "change size by" block that scales the size by 0.2 and adds 50 to it.

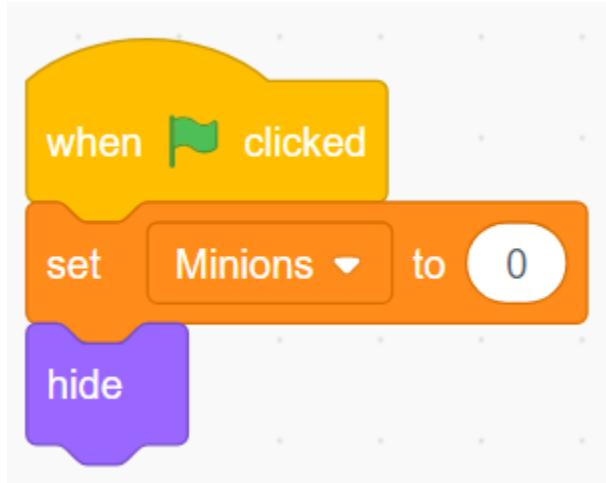
This code changes the costume and makes it animated:

```
define [change costume to :costume]
if [not (costume = costume name)] then
  switch costume to :costume
  set size to (60) %
```

The Scratch script defines a new function named "[change costume to :costume]" using a "define" hat block. This function takes one parameter, "costume". The script then uses an "if" control block with a condition "not (costume = costume name)". If the condition is true, it executes two actions: "switch costume to :costume" and "set size to (60) %".

Code For... THE BOSS

This is the starting code for the boss:



This code makes the boss show fire lasers, and move:

The Scratch script starts with a "when I receive Boss" hat event. It begins by moving the boss to position (0, 999) and switching its costume to "Level 1". It then sets the boss's Max Health to 50 times its current costume number (which is 5). The script contains two nested if statements. The first if statement checks if the costume number is 8; if so, it sets Max Health to 50. The second if statement checks if the costume number is 7; if so, it sets Max Health to 500. Both if statements have a long orange vertical slot below them. After these conditionals, the script sets the boss's Health to its Max Health and clears graphic effects. It then shows the boss. A "repeat until" loop begins, checking if the y position is less than 100. Inside the loop, the y position is changed by -1. The loop then ends with a "set boss speed x to 2" command. Following this, a "forever" loop begins, which changes the x position by the boss speed x. Inside this forever loop, there are two nested if statements. The first if statement checks if the x position is greater than 150; if so, the boss speed x is set to -2. The second if statement checks if the x position is less than -150; if so, the boss speed x is set back to 2.

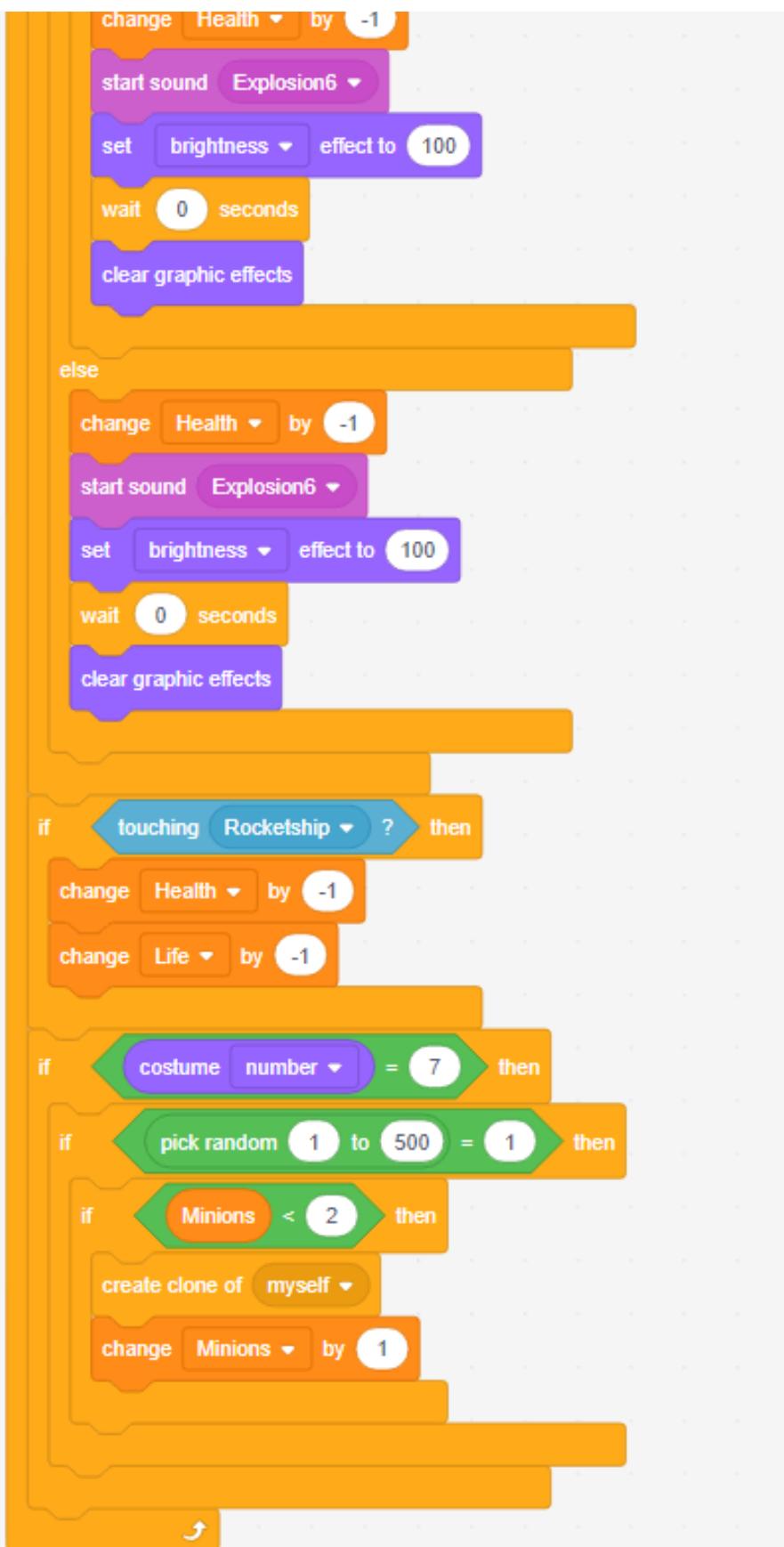
```
when I receive Boss
go to x: 0 y: 999
switch costume to [Level 1 v]
set [Max Health v] to [costume number * 50]
if [costume number = 8] then
  set [Max Health v] to [50]
end
if [costume number = 7] then
  set [Max Health v] to [500]
end
set [Health v] to [Max Health]
clear graphic effects
show
repeat until [y position < 100]
  change y by [-1]
end
set [boss speed x v] to [2]
forever
  change x by [boss speed x]
  if [x position > 150] then
    set [boss speed x v] to [-2]
  end
  if [x position < -150] then
    set [boss speed x v] to [2]
  end
```

```

if round pick random 1 to 100 / costume number ▾ = 1 then
  if costume number ▾ = 7 then
    if pick random 1 to 5 = 1 then
      add x position to Laser Positions ▾
      add y position to Laser Positions ▾
      create clone of Death Ray ▾
    else
      add x position to Laser Positions ▾
      add y position - 50 to Laser Positions ▾
      create clone of Enemy Laser ▾
      add x position + 50 to Laser Positions ▾
      add y position - 50 to Laser Positions ▾
      create clone of Enemy Laser ▾
      add x position + -50 to Laser Positions ▾
      add y position - 50 to Laser Positions ▾
      create clone of Enemy Laser ▾
    end
  end
end

if touching Laser ▾ ? then
  if costume number ▾ = 3 then
    if pick random 1 to 2 = 1 then
      change Health ▾ by -1
    end
  end
end

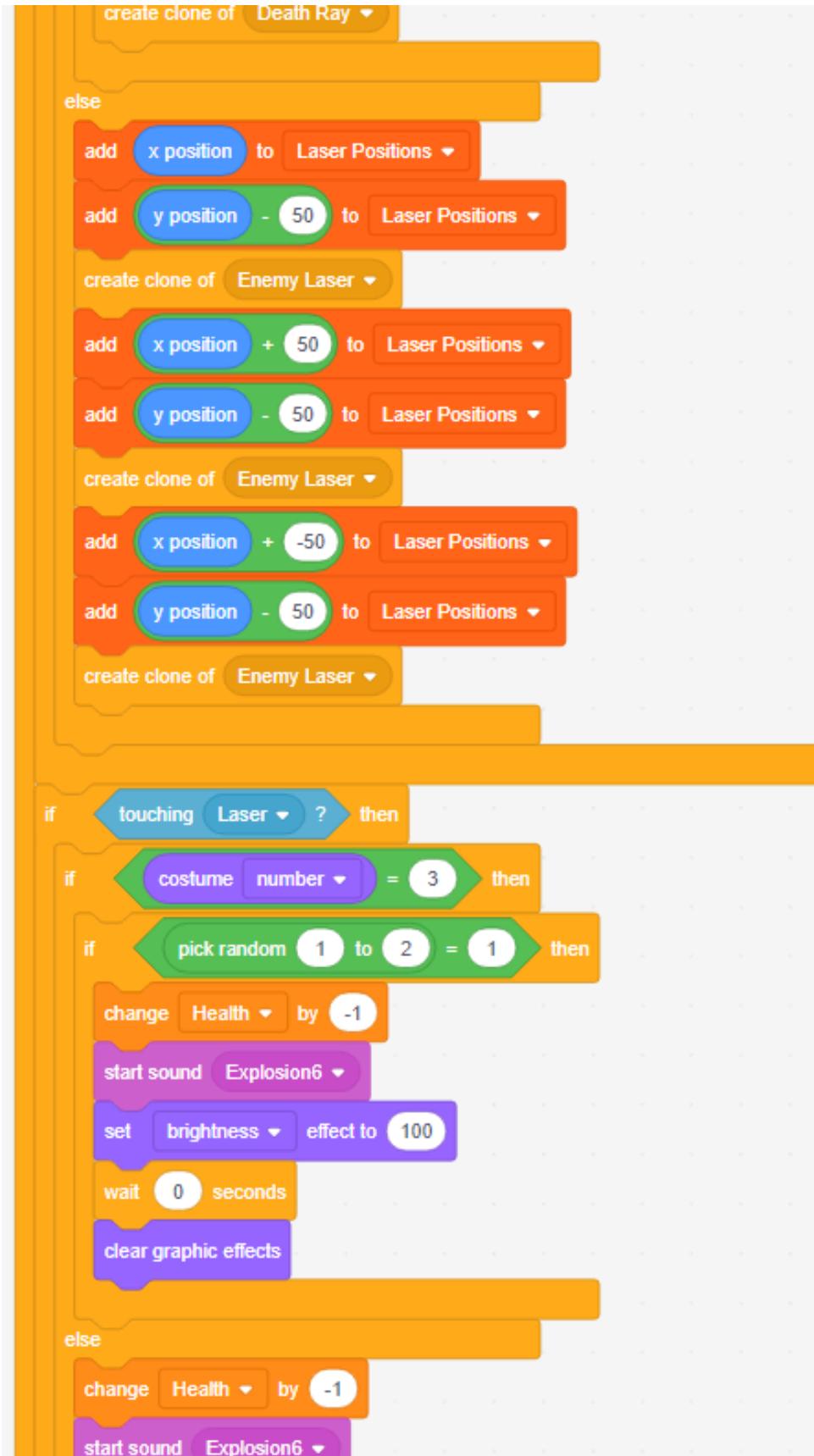
```



This code makes the minion show fire lasers, move, and jump:

The Scratch script starts with the "when I start as a clone" hat. It begins by switching the costume to "Final Boss Minion". It then sets the "Minion Speed" to 3. The script enters a "forever" loop. Inside the loop, the "Minion Speed" is increased by its current value. An "if" condition checks if the "x position" is greater than 150; if so, the speed is set to -4. Another "if" condition checks if the "x position" is less than -150; if so, the speed is set to 4. A nested "if" condition then checks if the "costume number" is 1. If it is, another nested "if" condition checks if the "costume number" is 7. If both are true, a series of actions are performed: adding the current "x position" and "y position" to lists named "Laser Positions", creating a clone of the "Death Ray" costume, and changing the costume to "Death Ray". If the "costume number" is not 7, the "else" branch adds the current "x position" and "y position minus 50" to the "Laser Positions" lists, and creates a clone of the "Enemy Laser" costume.

```
when I start as a clone
switch costume to [Final Boss Minion v]
go to [front v] [layer v]
set [Minion Speed v] to [3]
forever
  change [Minion Speed v] by [1]
  if [x position > 150] then
    set [Minion Speed v] to [-4]
  else
    if [x position < -150] then
      set [Minion Speed v] to [4]
    end
  end
  if [round (pick random [1] to [100]) / costume number = 1] then
    if [costume number = 7] then
      add (x position) to [Laser Positions v]
      add (y position) to [Laser Positions v]
      create clone of [Death Ray v]
    end
  end
  else
    add (x position) to [Laser Positions v]
    add [(y position) - 50] to [Laser Positions v]
    create clone of [Enemy Laser v]
  end
end
```

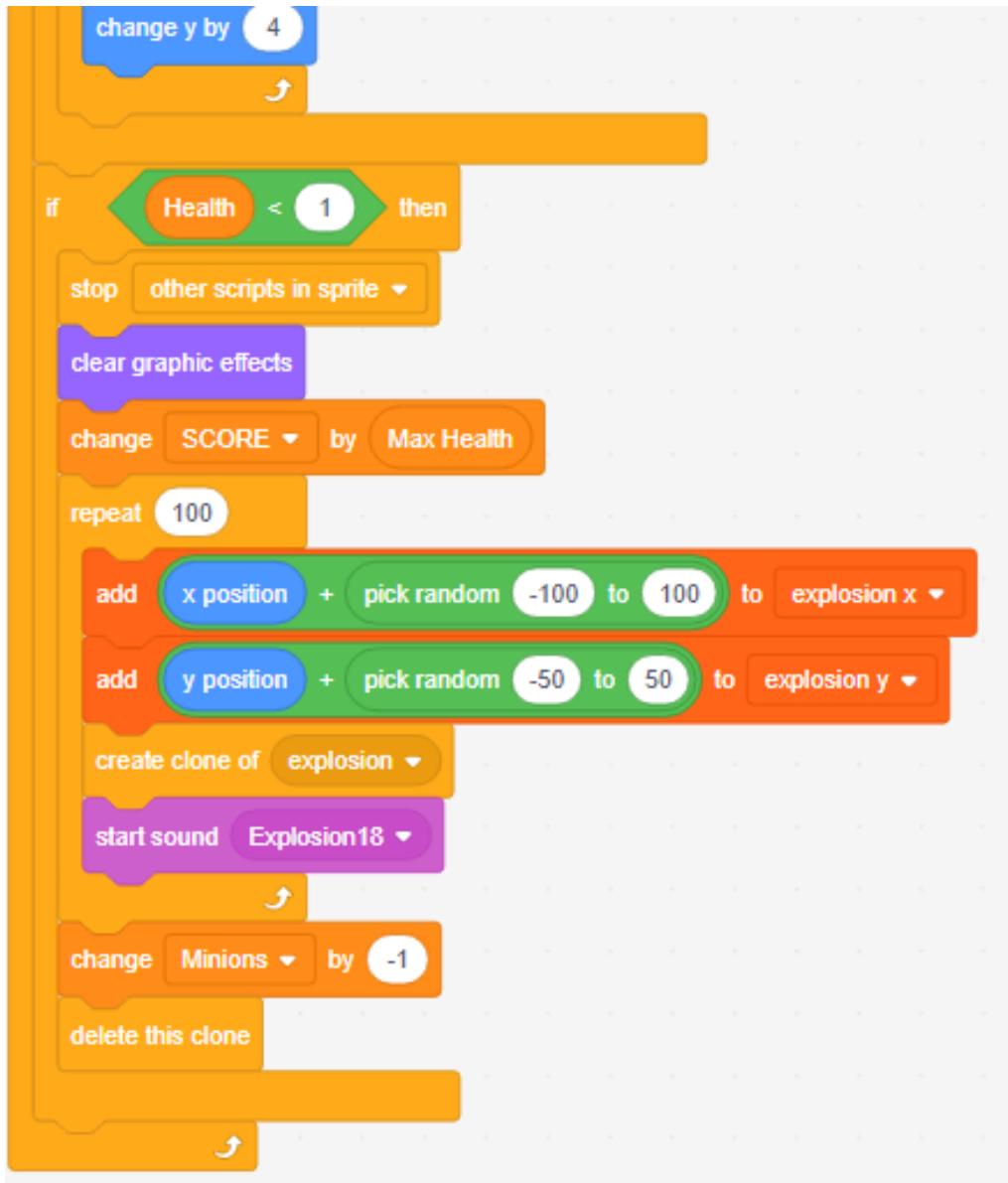


A Scratch script for a laser enemy. It starts with a `create clone of [Death Ray v]` hat block. Inside the loop, it checks if the laser has hit the player. If so, it checks if the costume number is 3. If yes, it changes health by -1, plays the `Explosion6` sound, sets brightness to 100, waits 0 seconds, and clears graphic effects. If costume number is not 3, it changes health by -1 and plays the `Explosion6` sound. Otherwise, it adds the current position to the laser positions list and creates clones of `Enemy Laser` at various offsets.

```
when green flag clicked
  [Death Ray v] hat
    [if touching [Laser v] ? then
      [if costume number = (3) then
        [if pick random (1) to (2) = (1) then
          change [Health v] by (-1)
          start sound [Explosion6 v]
          set [brightness v] effect to (100)
          wait (0) seconds
          clear graphic effects
        end
      end
      else
        change [Health v] by (-1)
        start sound [Explosion6 v]
      end
      add [x position] to [Laser Positions v]
      add ([y position] - (50)) to [Laser Positions v]
      create clone of [Enemy Laser v]
      add ([x position] + (50)) to [Laser Positions v]
      add ([y position] - (50)) to [Laser Positions v]
      create clone of [Enemy Laser v]
      add ([x position] + (-50)) to [Laser Positions v]
      add ([y position] - (50)) to [Laser Positions v]
      create clone of [Enemy Laser v]
    end
  end
end
```

This Scratch script performs several actions based on collision detection:

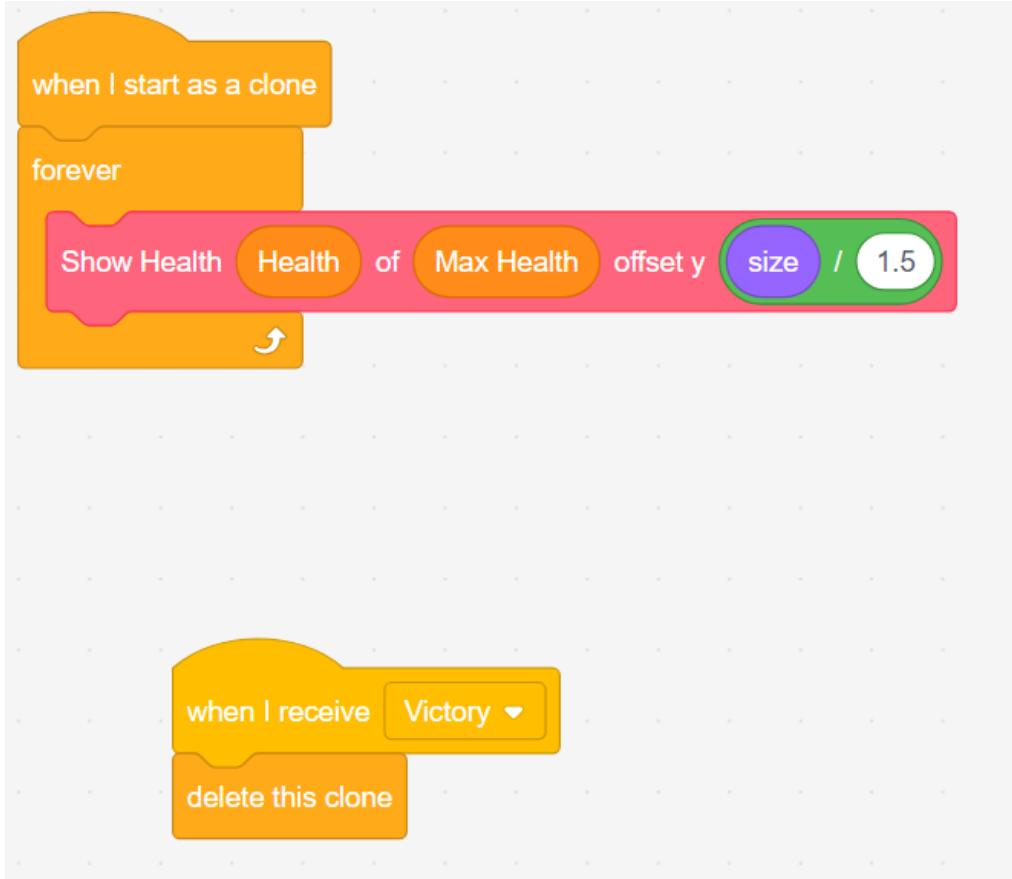
- If the player's costume number is 7:
 - If a random value between 1 and 1000 is 1:
 - Create a clone of the player.
 - If a random value between 1 and 200 is 1:
 - Repeat 10 [Change y by -20]
 - Repeat 50 [Change y by 4]
- If the player is touching a Rocketship:
 - Decrease Health by 1.
 - Decrease Life by 1.
- Else:
 - Start sound "Explosion6".
 - Set brightness effect to 100.
 - Wait 0 seconds.
 - Clear graphic effects.



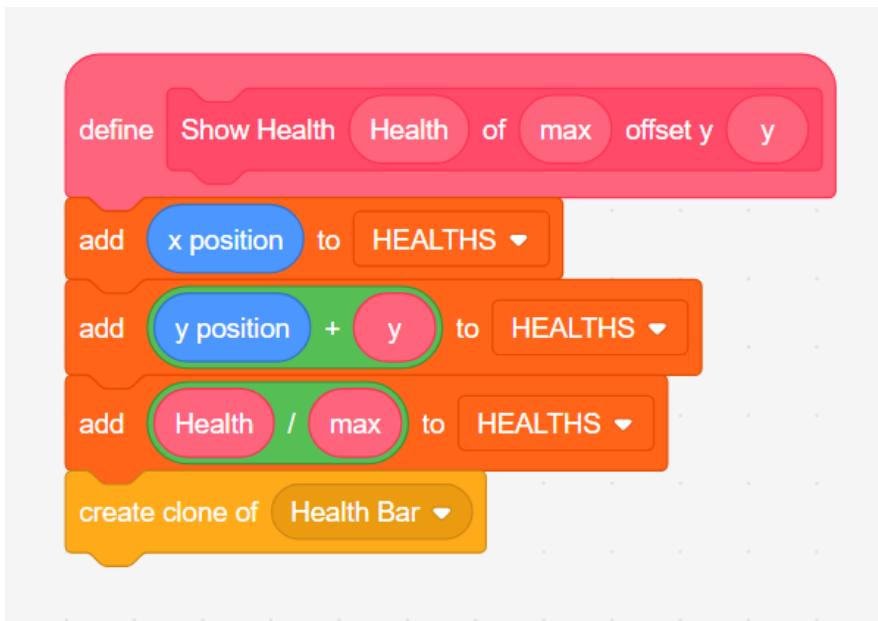
This code makes the boss be able to be hit by lasers:

```
when I receive Boss
forever
  Show Health [Health] of [Max Health] offset y [size / 1.5]
  if [Health < 1] then
    stop other scripts in sprite
    clear graphic effects
    change [SCORE v] by [Max Health]
    if [costume number] = [7] then
      broadcast [Victory v]
    end
    repeat (100)
      add [x position + pick random (-100) to 100] to [explosion x v]
      add [y position + pick random (-50) to 50] to [explosion y v]
      create clone of [explosion v]
      start sound [Explosion18 v]
    end
    hide
    if [not costume number] = [7] then
      broadcast [Next Level v]
    end
    stop this script
  end
```

This code shows the minions health and this code gets rid of the minions when you beat the game:

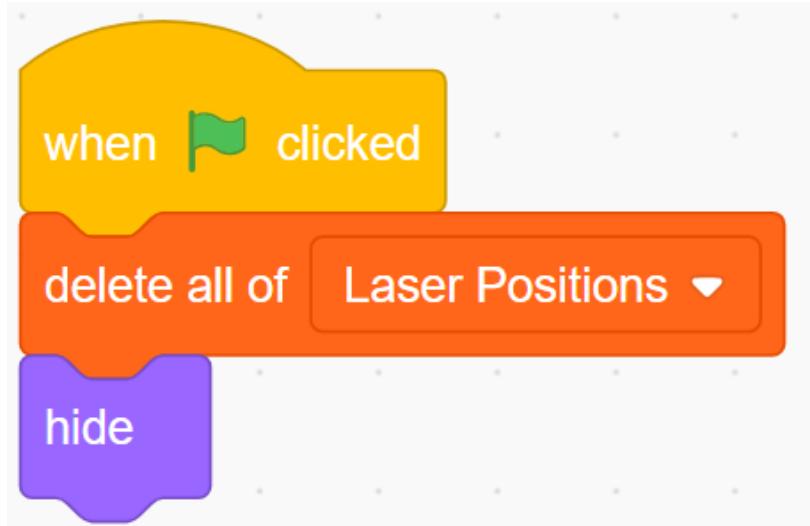


This code shows the minions health:

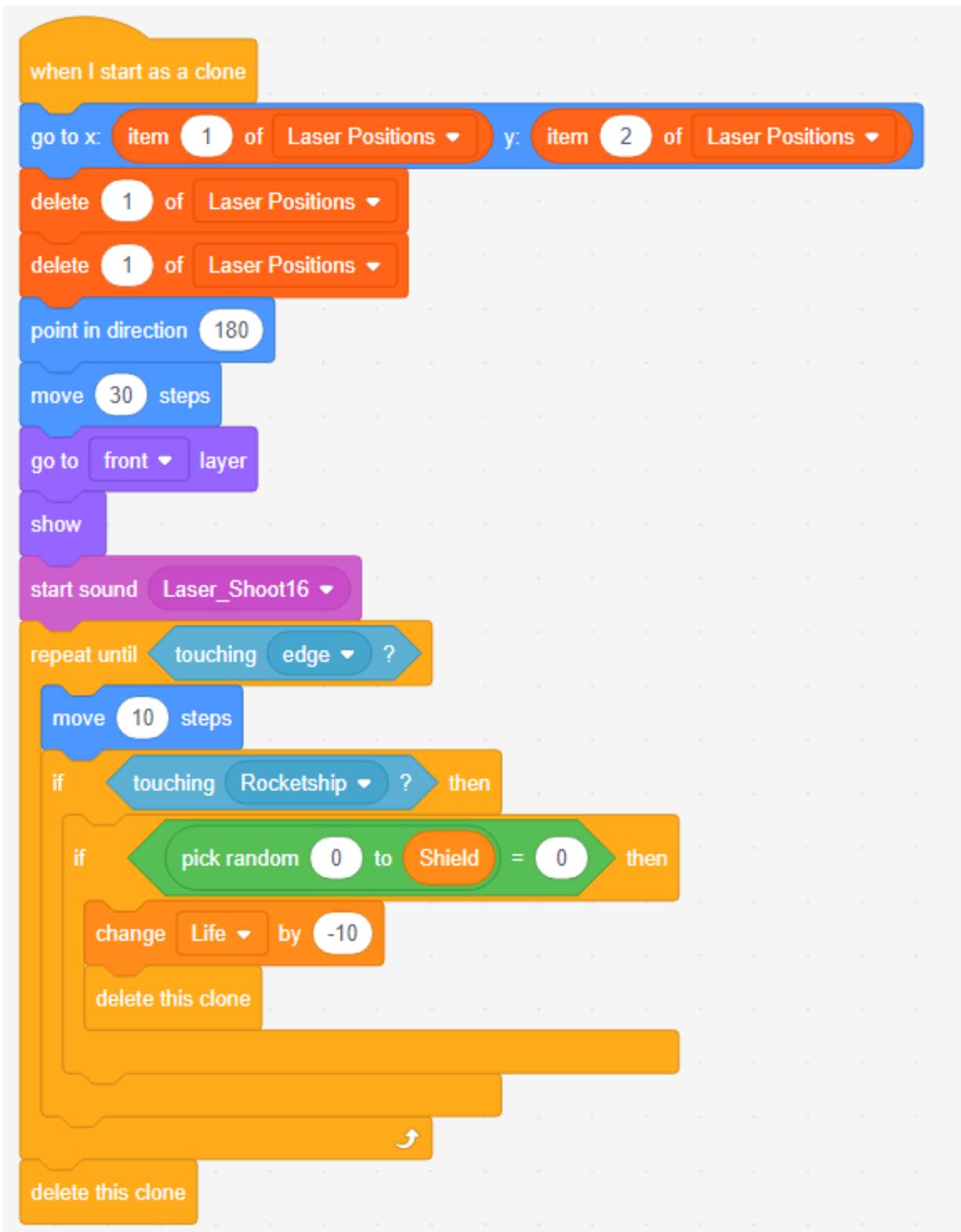


Code For The Enemy Laser

This is just the starting code for the enemy laser:



This code makes the enemy laser show and move down:



Code For The Enemy Spawn

This code spawns the enemies:

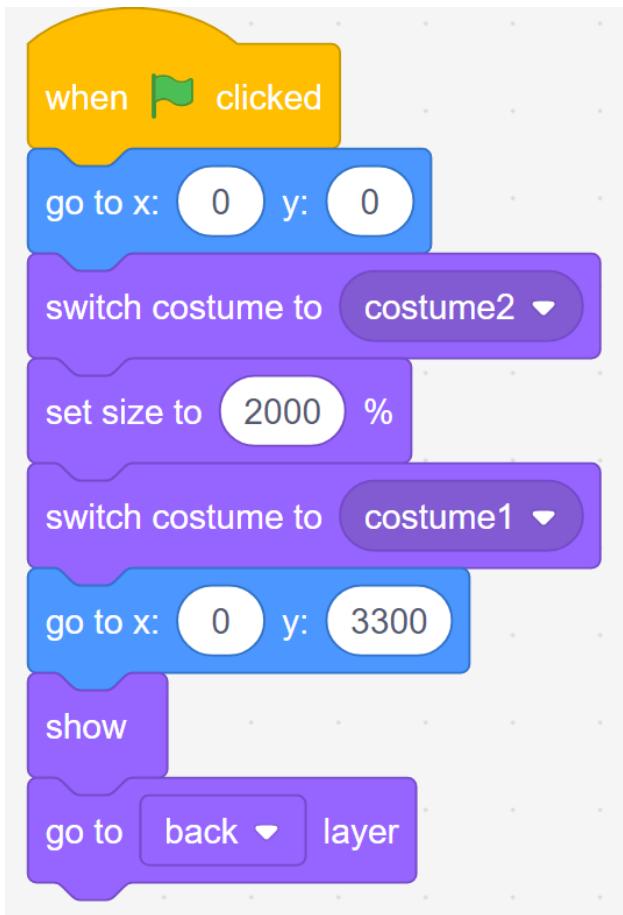


This code stops more enemies from spawning:

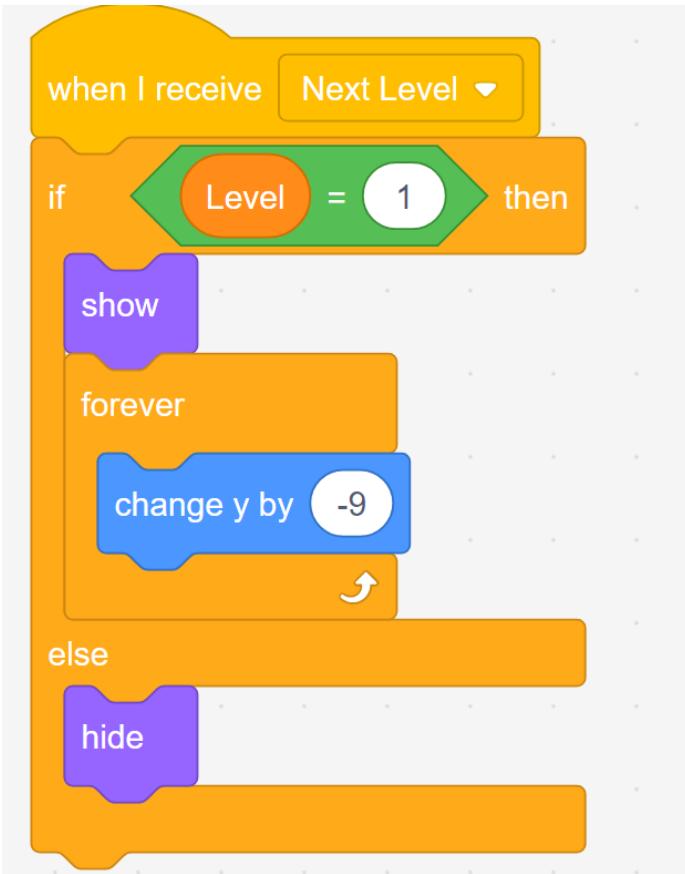


Code For The Earth

This is the starting code for the ground:

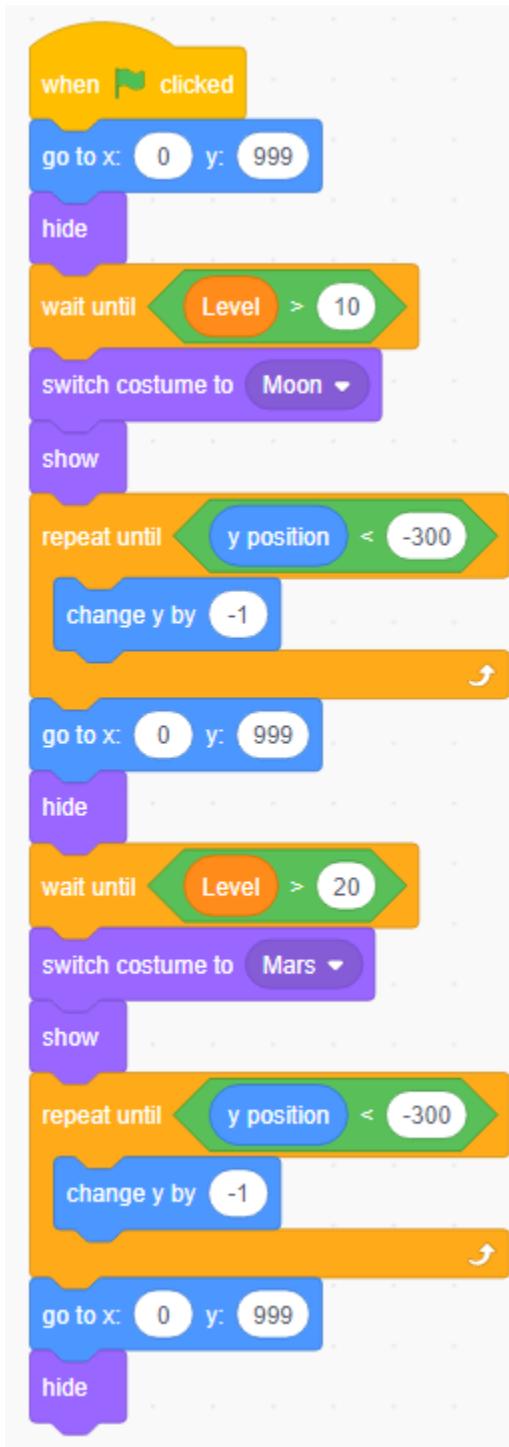


This code makes the ground move down when you are flying:



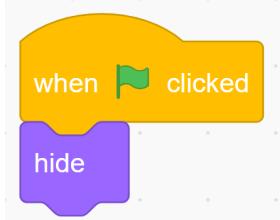
Code For The Planets

This code makes planets or moons appear occasionally after 2 boss fights:

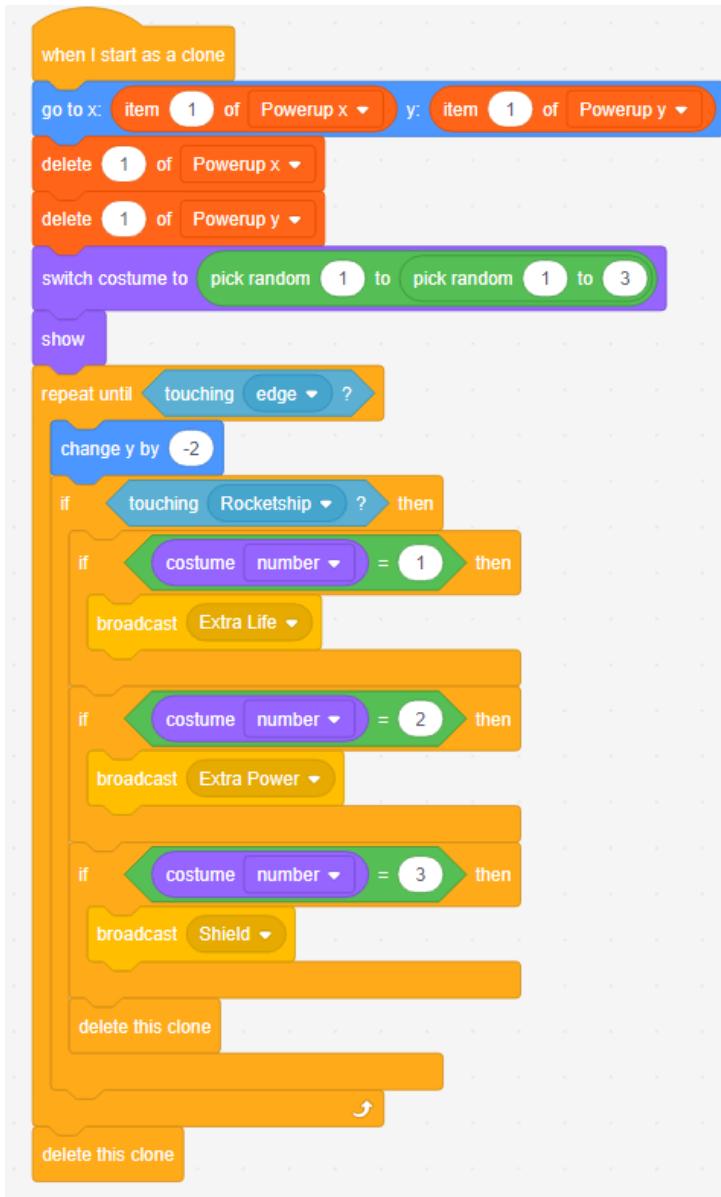


Code For The Power-Ups

This code makes the original powerup sprite show:

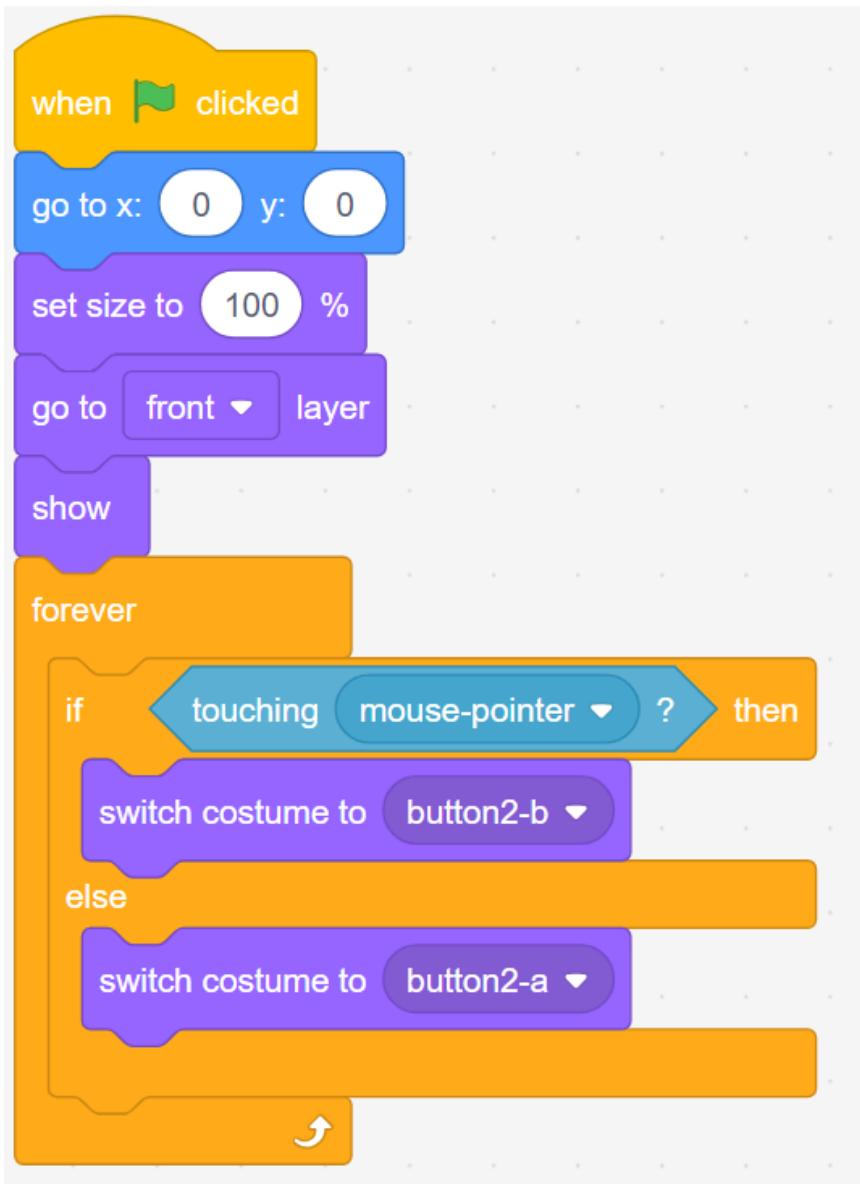


This code makes the powerups move and be able to be collected:

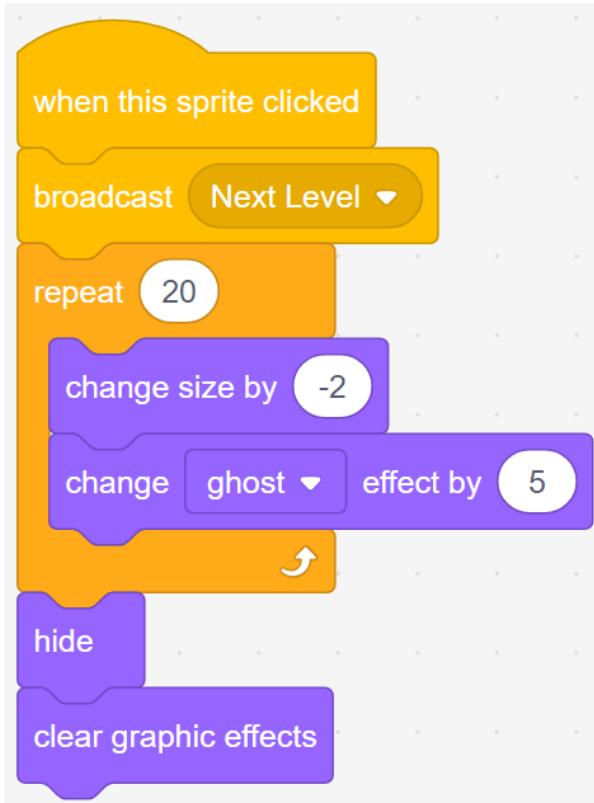


Code For The Play Button

This code makes the play button show at the beginning and change colors when it touches the mouse pointer:

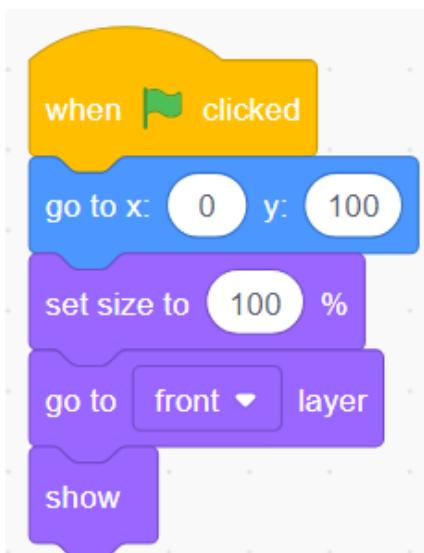


This code makes the play button disappear after the game starts:

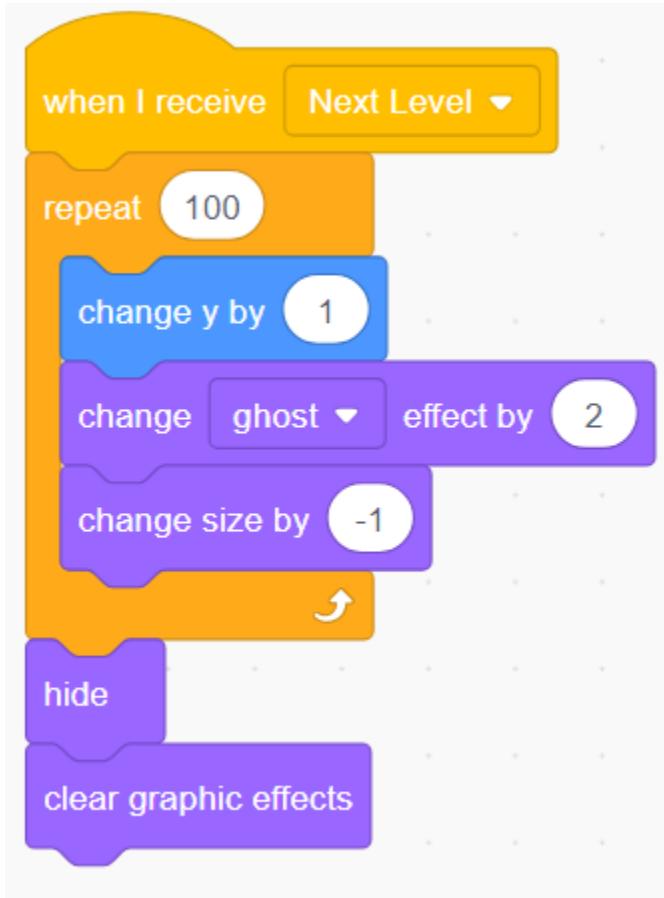


Code For The Title

This makes the title show at the beginning of the game:

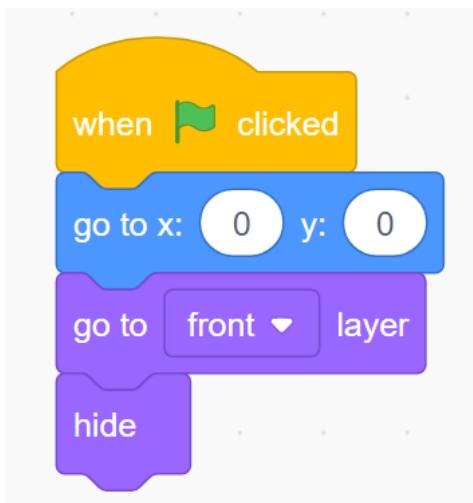


This code makes the title disappear after the play button is pressed:

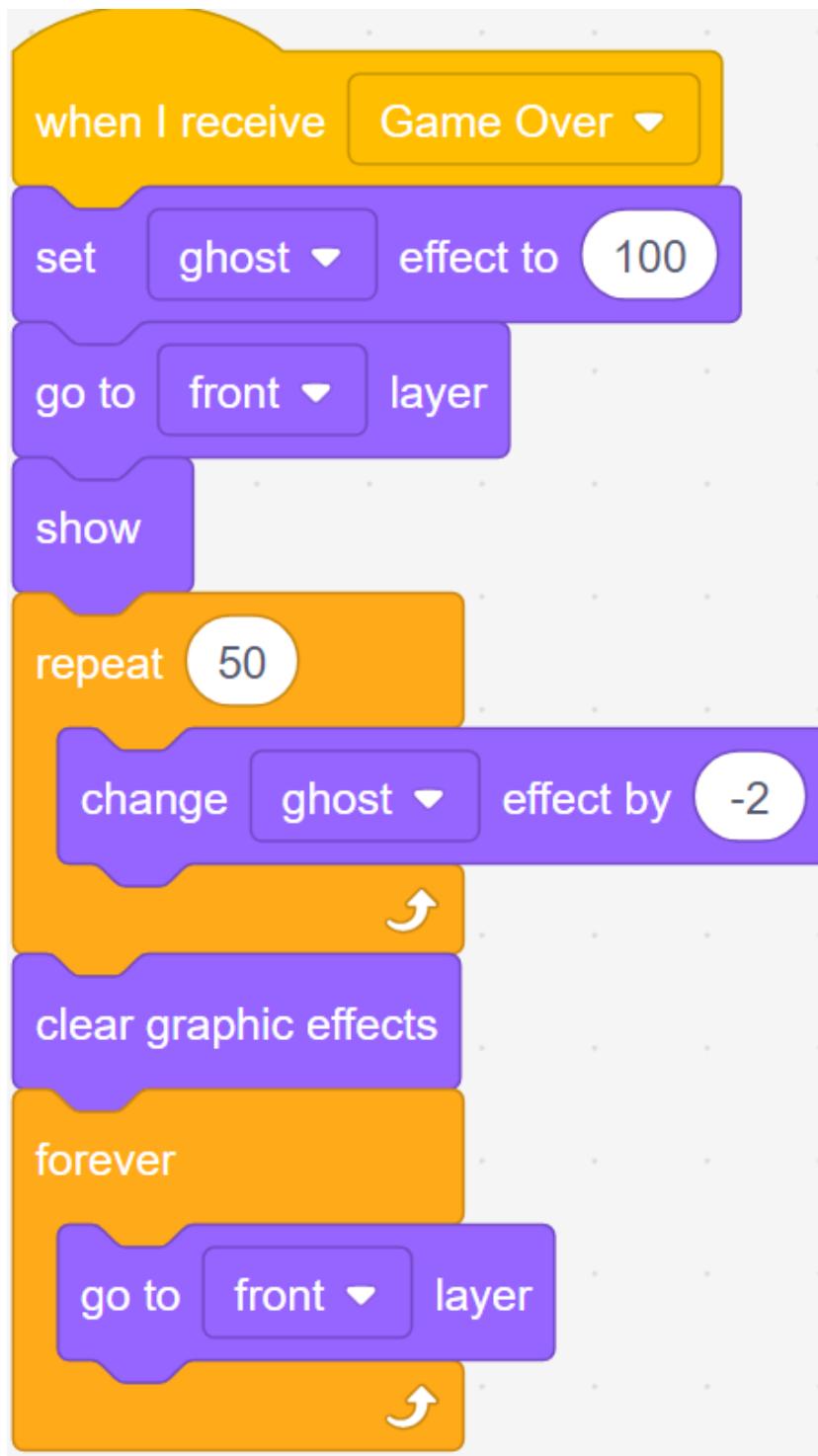


Code For The Game Over

This code makes the game over hide when the green flag is pressed:

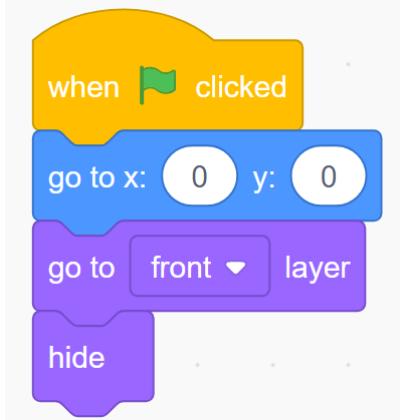


This makes the game over show after you die:

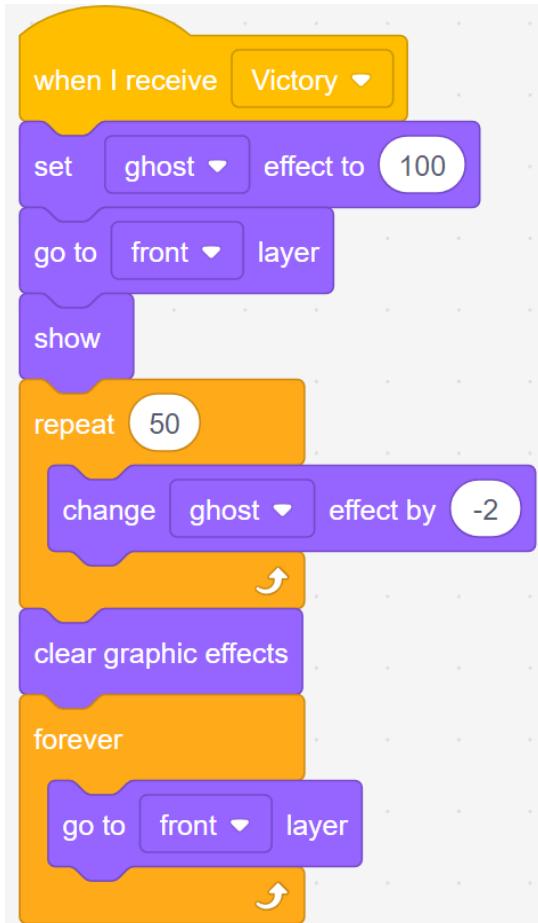


Code For The Victory

This code makes the victory hide when the green flag is pressed:

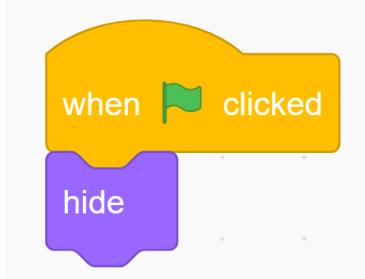


This makes the victory show after you win:

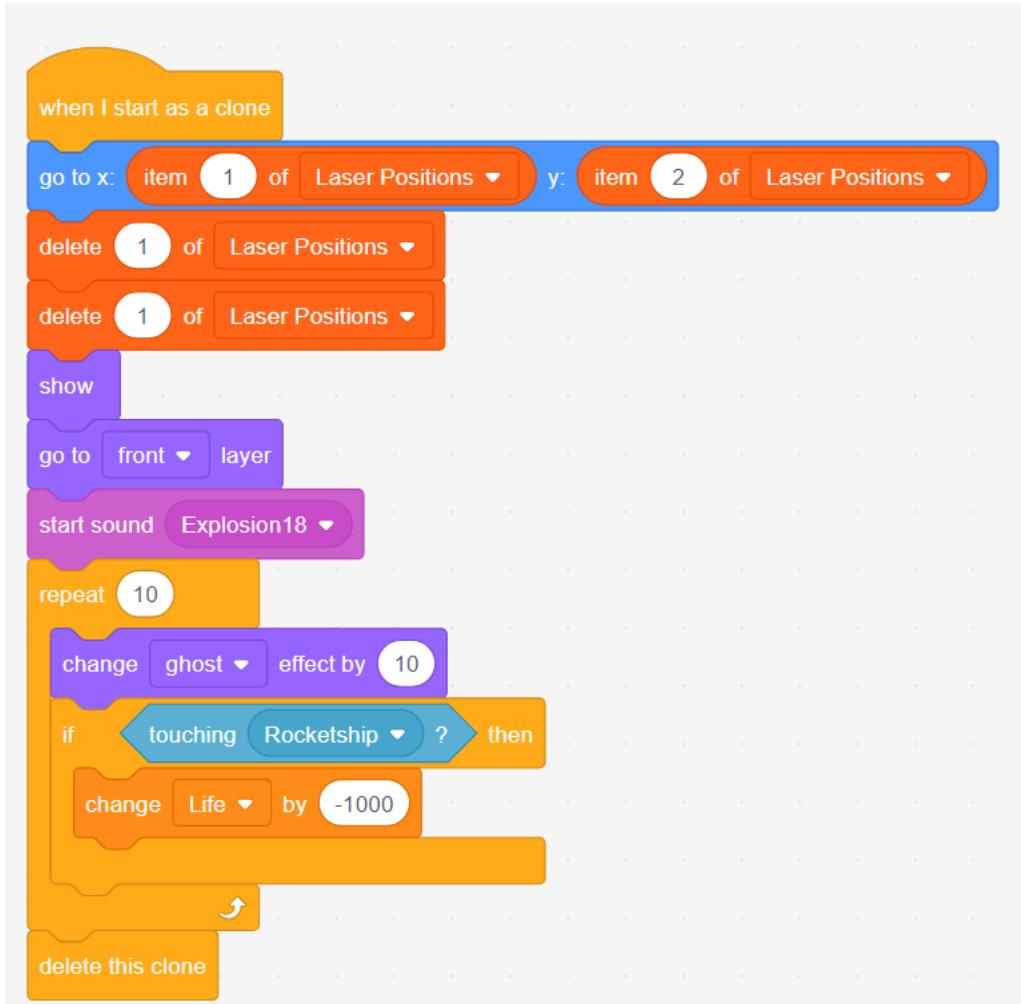


Code For The Death Ray

This is just the code that makes the main sprite hidden:

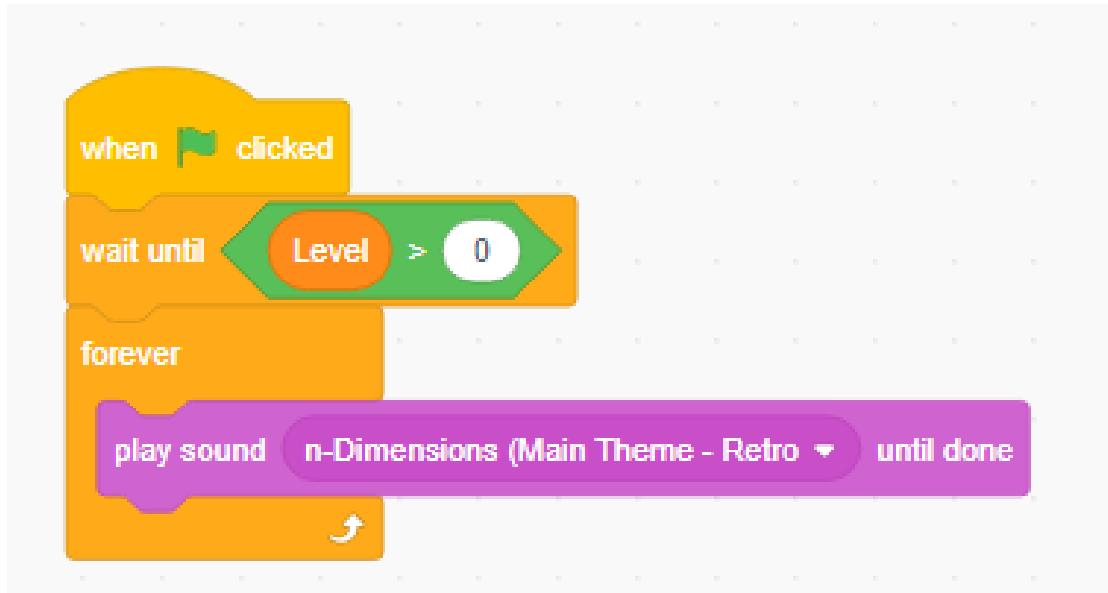


This is the code that makes the death ray appear:



Code For The Background

This code makes the background music play:



Reference

- Ciocîrlan, D. (2022, May). *Advanced scratch programming for kids*. Udemy.
<https://www.udemy.com/course/advanced-scratch-programming/>
- datsvs. (2018, April 25). *Moon*. Purepng. <https://purepng.com/photo/5841/nature-moon>
- Death Star transparent background PNG cliparts free download | HiClipart*. (n.d.).
<https://www.hiclipart.com/search?clipart=death+Star>
- griffpatch_tutor. (2021, June 28). *Lemonoid Assets (Part 1 & 2)*. Scratch.
<https://scratch.mit.edu/projects/548866331/>
- Scratch - imagine, program, share*. (n.d.). <https://scratch.mit.edu/>